
Universidade de Aveiro
2009
Departamento de Electrónica, Telecomunicações e Informática

Miguel Ângelo Lopes Gaspar Veiga

Simulação de Redes MPLS: Uma Perspectiva Pedagógica

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Nogueira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho aos meus Pais, à minha irmã, à minha sobrinha e aos meus amigos.

o júri

Presidente

Prof. Dr. A. Oliveira Duarte
Professor Catedrático da Universidade de Aveiro

Prof. Dr. Joel José Puga Coelho Rodrigues
Professor Auxiliar da Universidade da Beira Interior

Prof. Dr. António Manuel Duarte Nogueira
Professor Auxiliar da Universidade de Aveiro

Prof. Dr. Paulo Jorge Salvador Serra Ferreira
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Gostaria de agradecer em primeiro lugar ao meu orientador Prof. Dr. António Nogueira pela disponibilidade e paciência que teve comigo durante a realização deste trabalho. Quero também agradecer aos meus Pais, irmã e Padrinho pela ajuda que me prestaram.
Agradeço também aos meus amigos, Daniel Albuquerque, Hélio Edgar Araújo, Rodolphe Marques e Ricardo Filipe pela ajuda imprescindível prestada.
Agradeço à familia Darling, Fernando, Ricardo, Nelson Machado, Daniela, Paula e Tia Maria, à Patricia Pereira e a todas as Pipocas (elas sabem quem são) pela paciência que tiveram para me aturar durante os últimos meses e pelo apoio dado dia após dia.
A todos o meu muito obrigado pela ajuda, apoio e incentivo prestado à minha pessoa.

Palavras-chave

Redes MPLS,Rotas Estáticas, Rotas Dinâmicas, Balanceamento de Carga, VPNs, MPLS-TE.

Resumo

Este trabalho propõe um conjunto de experiências laboratoriais desenvolvidas no sentido de elucidar os principais conceitos da tecnologia MPLS (Multiprotocol Label Switching). Dada a importância crescente desta tecnologia no contexto das redes IP, torna-se importante que ela seja leccionada nas disciplinas de Redes dos diversos Mestrados Integrados do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e, nesse sentido, foram idealizadas diversas experiências laboratoriais que permitem compreender, de uma forma simples, os principais mecanismos de funcionamento do MPLS. Este protocolo permite optimizar o envio de pacotes através de uma rede mediante a utilização de rótulos, isto é, permite controlar a forma como o tráfego flui através da rede por forma a optimizar o seu desempenho e a utilização dos seus recursos. O MPLS foi desenvolvido com o propósito de criar uma solução normalizada que funcione em conjunto com o protocolo IP, capaz de expedir pacotes a um ritmo elevado e que possua mecanismos de gestão de recursos da rede que permitam o tratamento diferenciado de fluxos de tráfego com requisitos variados de qualidade de serviço.

O conjunto de experiências desenvolvidas teve em atenção o facto de muitas das aulas práticas de Redes leccionadas no DETI terem lugar em laboratórios de redes onde os alunos realizam os trabalhos em grupos de duas pessoas e dispõem de um número limitado (embora grande) de equipamentos de rede. Assim, as experiências propostas neste trabalho utilizam, de uma forma geral, apenas 3 ou 4 routers, o que permite a sua realização no ambiente laboratorial das aulas práticas de redes. A opção pela utilização do simulador GNS3 teve a ver com a possibilidade de, por um lado, optimizar o tempo disponível para a realização do trabalho e contornar a limitação dos espaços físicos disponibilizados para a sua realização e, por outro lado, constituir um ambiente versátil e que não obriga à utilização de equipamento real.

keywords MPLS Networks, Estatic Routes, Dynamic Routes, Load-Share,VPNs, MPLS-TE.

abstract This work proposes a set of laboratorial experiments developed in a sense to elucidate the main concepts of the MPLS (Multiprotocol Label Switching) technology. Given the growing importance of this technology in the IP networking context, it is important that it can be conveniently taught in the various networking disciplines of the Integrated Master Courses of the Department of Electronics, Telecommunications and Informatics of the University of Aveiro. Therefore, several laboratory experiments were idealized in order to allow students to understand, in a simple way, the main MPLS functioning mechanisms. This protocol allows optimizing the transmission of packets through the network by using of labels, that is, it is possible to control how traffic flows through the network in order to optimize its performance and resource usage. The MPLS was developed in order to create a standardized solution that works in conjunction with the IP protocol, is able to send packets at a very high rate and has mechanisms that facilitate the management of the network resources, allowing a differentiated treatment of the traffic flows that have different quality of service (QoS) requirements.
The set of experiments was developed having in mind the fact that many of the practical networking classes that are given at DETI take place in network laboratories where students are divided in groups of two elements and each group has a limited (although usually high) number of network equipments. Therefore, the proposed experiments use, in general, 3 or 4 routers, which make them perfectly executable in the lab environment of our practical classes. The option of using the GNS3 network simulator was based on the fact that this environment allows, on one hand, maximizing the limited time that is available to carry out the work and overcoming the limited physical space that is available for its completion and, on the other hand, providing a versatile environment that does not require the use of any physical networking equipment.

Conteúdo

Lista de Figuras

Lista de Tabelas

1. Introdução

Nas últimas décadas, as redes de telecomunicações sofreram grandes transformações. Os avanços tecnológicos permitiram o desenvolvimento de sistemas de transmissão de alto débito. Simultaneamente, o consequente aumento da capacidade de processamento e armazenamento dos equipamentos terminais e o desenvolvimento de novos métodos de processamento digital de som, imagem e vídeo conduziram ao aparecimento de novas aplicações telemáticas, que exigem das redes de telecomunicações uma maior capacidade e flexibilidade na transmissão da informação.

No final dos anos 90 foi introduzido o protocolo MPLS (Multiprotocol Label Switching), que permite controlar a forma como o tráfego flui através da rede IP por forma a optimizar o desempenho e a utilização dos recursos da rede. A essência do MPLS é a utilização de um rótulo de tamanho fixo, que funciona como abreviatura do endereço IP do pacote, e que é utilizado no processo de tomada de decisões de encaminhamento do pacote. O MPLS pode ser utilizado com as versões 4 e 6 do protocolo IP, não estando também limitado a nenhuma tecnologia específica ao nível da ligação de dados. O MPLS veio então fornecer soluções que permitem aumentar a eficiência do encaminhamento de pacotes, contemplar a diferenciação entre os serviços QoS e CoS (classes de serviços), facilitar o crescimento das redes IP, integrar a rede IP com as tecnologias de nível 2 e auxiliar na implementação de VPNs (Redes Privadas Virtuais) com capacidade de engenharia de tráfego.

Neste trabalho é feita uma abordagem, de um ponto de vista pedagógico, de algumas das características importantes da tecnologia MPLS. Nesse sentido, foram desenvolvidas pequenas experiências passíveis de serem realizadas nas aulas práticas de Redes pelos alunos do DETI-UA (Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro) pertencentes aos Mestrados Integrados em Engenharia Electrónica, Telecomunicações e Informática e Engenharia de Computadores e Telemática, de modo a permitir-lhes adquirir uma melhor compreensão do funcionamento de alguss dos mecanismos peculiares do MPLS.

As experiências concebidas podem ser realizadas em laboratório, recorrendo ao material disponível, ou então recorrendo a simuladores que permitem simular ambientes

reais de redes IP. Neste trabalho, optou-se por utilizar o simulador GNS3 (www.gns3.net/) no sentido de, por um lado, optimizar o tempo disponível para a realização do trabalho e contornar a limitação dos espaços físicos disponibilizados para a sua realização e, por outro lado, utilizar um ambiente versátil e que não obriga à utilização de equipamento real. No entanto, todas as experiências concebidas podem ser facilmente realizadas nas disciplinas de Redes dos Mestrados Integrados do DETI.

Esta dissertação divide-se em mais 8 capítulos. No Capítulo 2, Estado da Arte, descrevem-se todas as características e principais conceitos do MPLS e das tecnologias que giram à sua volta: os elementos que constituem uma rede MPLS, os mecanismos de distribuição de rótulos, as aplicações do MPLS, a Engenharia de Tráfego em MPLS, o conceito de Redes Privadas Virtuais (VPNs), a implementação de túneis e os simuladores de rede existente no mercado.

A partir do Capítulo 3 inicia-se a descrição das experiências realizadas. No Capítulo 3 são referidos os conceitos básicos de configuração do MPLS, incluindo a configuração de túneis. Neste Capítulo, ainda são apresentadas diversas experiências envolvendo a criação de túneis MPLS, focando o efeito de algumas das suas características: a prioridade, o parâmetro *Path Option*, segundo diversas abordagens, e o parâmetro *Load-Share*.

No Capítulo 4 é estudada a implementação de VPNs sobre MPLS.

Por último, no Capítulo 5, são apresentadas algumas das conclusões mais importantes recolhidas ao longo deste trabalho, assim como algumas ideias para trabalho futuro.

2. Estado da Arte

No final dos anos 90, foi introduzida a tecnologia MPLS, que permite controlar a forma como o tráfego flui através de redes IP por forma a optimizar o desempenho da rede e a utilização dos seus recursos. O MPLS foi desenvolvido pelo Internet Engineering Task Force (IETF) com o propósito de criar uma solução normalizada que funcione em conjunto com o protocolo IP, capaz de expedir pacotes a um ritmo superior e que possua mecanismos de gestão dos recursos da rede por forma a permitir o tratamento diferenciado dos fluxos de tráfego com variados requisitos de qualidade de serviço (QoS), incluindo a reserva de recursos [**1**]. A sua essência é a utilização de um rótulo de tamanho fixo, que funciona como uma abreviatura de endereço IP do pacote e que é utilizado para a tomada de decisões de encaminhamento IP.

As três principais promessas do MPLS foram a Engenharia de Tráfego, a garantia de Qualidade de Serviço e possibilidade de implementar facilmente Redes Privadas Virtuais (Virtual Private Networks -VPNs).

No Encaminhamento IP Tradicional, as decisões de expedição de pacotes requerem uma pesquisa do tipo *longest match*, que compara o endereço de destino do pacote com cada uma das entradas na tabela de encaminhamento, procedimento repetido em cada nó do percurso, desde a origem ao destino. Embora logicamente distintos, os planos de controlo e expedição estão fortemente ligados.

Em MPLS, a utilização do rótulo permite separar os planos de controlo e expedição: o plano de controlo utiliza protocolos de encaminhamento, como o OSPF, para construir e actualizar as tabelas de encaminhamento dos routers. A ligação entre o plano de controlo e o plano de expedição é feito através da criação de *Forwarding Equivalence Classes* (FEC), que mapeiam as entradas na tabela de expedição com os rótulos a atribuir ao pacote. Este mapeamento é realizado localmente em cada um dos *Label Switching Routers* (LSRs), sendo necessário publicitar esta informação aos LSRs vizinhos através de um protocolo apropriado. Assim, ao receber um pacote, o LSR usa o rótulo para indexar a tabela de expedição e obter a informação necessária para o encaminhamento do pacote – interface de saída e novo rótulo. Este processo pode ser realizado por hardware dedicado, aumentando bastante a velocidade de comutação de pacotes.

2.1. Elementos de uma Rede MPLS

Os routers que formam as redes MPLS são denominados de Label Switching Routers (LSR) ou Label Edge Routers (LER), dependendo do seu papel. Um LSR é um router do núcleo da rede MPLS, participa no estabelecimento de LSPs usando protocolos de distribuição de rótulos e é capaz de efectuar a expedição de pacotes rotulados a ritmos elevados, bem como encaminhamento IP convencional. Na fronteira da rede, os LERs implementam políticas de gestão e acesso determinadas pelo administrador da rede e efectuam a classificação e agregação de tráfego, inserindo (routers de entrada), ou removendo (routers de saida) rótulos nos pacotes. Utiliza-se a designação de LSR para referir ambos os tipos de routers MPLS.

Um LSP (*Label Switched Path*) é o percurso que os pacotes percorrem entre o nó de ingresso e o nó de engresso. Uma FEC (*Forwarding Equivalence Class*) é um conjunto de pacotes tratados, para efeitos de expedição, de uma forma equivalente. A classificação de um pacote numa FEC é feita através do processamento dos campos do cabeçalho: requisitos de QoS, tipo de aplicação, identificador da VPN, sub-rede de origem ou destino, ou grupo de IP multicast, e resulta na atribuição de um rótulo apropriado ao pacote.

É possível atribuir vários FECs ao mesmo LSP, e vários LSPs à mesma FEC, simplificando assim a agregação de fluxos multicast. Outra grande vantagem é a possibilidade de configurar explicitamente os LSPs atribuídos a cada FEC – é possível impor, de forma administrativa ou através de encaminhamento baseado em restrições, os percursos de cada fluxo de tráfego a encaminhar. Estas três características permitem um controlo preciso do tráfego na rede, o que torna as redes eficientes e os serviços suportados mais previsíveis e permite realizar, eficientemente, Engenharia de Tráfego.

Os rótulos são associados a FECs como resultado de um evento que indica a necessidade dessa associação. Estes eventos podem ser de dois tipos:

- *Data Driven*: a associação é efectuada quando chega a um LSR tráfego identificado como sendo candidato a *Label Switching*. As associações de rótulos a FECs só são estabelecidas quando necessário, resultando num menor número de entradas na tabela de expedição;

- *Control Driven*: as associações são feitas como consequência da actividade do plano de controlo e são independentes da informação a transportar. A escalabilidade deste método é superior à do Data-Driven, sendo por esta razão usada em MPLS.

O rótulo inicialmente inserido no pacote determina todo o seu percurso no domínio MPLS. Pode ser encapsulado de diversas formas, ou na camada de ligação lógica (ATM, Frame-Relay, VCI/VPI) ou encaixado num pequeno cabeçalho entre o cabeçalho da camada de ligação lógica e o cabeçalho da camada de rede (camadas 2 e 3, respectivamente, do modelo de referência OSI). Estas técnicas permitem que o MPLS possa ser suportado por qualquer protocolo e qualquer tecnologia da camada de ligação. O rótulo MPLS é constituído pelos seguintes campos:

- Rótulo: valor do rótulo MPLS;

- *Class of Service* (CoS): determina a forma como o pacote é tratado nas filas de espera dos routers de rede;

- *Stack* (S): permite a hierarquização de rótulos;

- *Time To Live* (TTL): permite a funcionalidade TTL IP convencional.

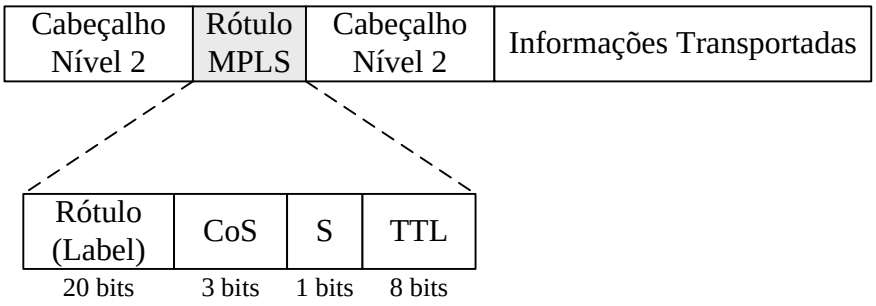


Figura 1 – Estrutura e encapsulamento do cabeçalho MPLS

2.2. Distribuição de Rótulos

Cada LSR atribui a um LSP um rótulo próprio. Assim, o router a montante do fluxo de tráfego tem de saber qual o rótulo que o router a jusante usa para identificar esse LSP. É então necessário que os LSRs distribuam informação sobre as associações rótulo/FEC efectuadas. Esta informação pode ser distribuída de duas formas:

- através do uso de um protocolo de distribuição de rótulos: o MPLS *Working Group* do IEFT definiu um novo protocolo – *Label Distribution Protocol* (LDP) –

especificamente para a distribuição da informação de mapeamento de rótulos. Este protocolo suporta a alocação de rótulos dos tipos *Data-Driven* e *Control-Driven*. A desvantagem do uso deste protocolo é o aumento da sua complexidade com os protocolos de encaminhamento. Posteriormente, o LDP foi alterado por forma a suportar encaminhamento com restrições.

- através de p*iggybacking* num protocolo de encaminhamento: a informação de associação de rótulos pode ser adicionada aos protocolos de encaminhamento tradicionais. Este método garante consistência na informação de expedição e evita o uso de outro protocolo. Infelizmente, nem todos os protocolos podem ser facilmente alterados para suportar *piggybacking*, pelo que este método não é uma solução completa para a distribuição de rótulos (BGP e RSVP foram alterados para suportar este protocolo) [**1**].

2.3. Expedição de pacotes em redes MPLS

Na fronteira do domínio MPLS são necessários routers de alto desempenho cuja função é implementar políticas de gestão e acesso, efectuar a agregação de tráfego, classificar os pacotes e aplicar (routers de ingresso), ou remover (routers de egresso) rótulos. A classificação de um pacote pode ser feita com base em informação tão diversa como o endereço de origem, endereço de destino, requisitos de QoS, aplicações de destino, identificador VPN ou estado actual da rede.

Os LSRs do núcleo da rede tomam a decisão de encaminhamento dos pacotes com base no seu rótulo: quando recebe um pacote, o LSR utiliza o rótulo como índice de tabela de expedição, obtendo assim toda a informação de que necessita para enviar um pacote – novo rótulo e LSR de destino.

Uma vez que o mapeamento entre rótulos é constante em cada LSR, o caminho de um pacote dentro da rede MPLS é determinado pelo seu rótulo inicial.

2.4. Encaminhamento baseado em restrições (EBR)

Nos casos dos protocolos de encaminhamento tradicionais, o administrador de rede define o custo de cada uma das ligações da rede. Para determinar o percurso entre um par origem-

destino, o protocolo de encaminhamento escolhe o percurso que minimiza a soma dos custos das ligações que pertencem ao caminho escolhido. Esta forma de determinar os percursos na rede é suficiente para conseguir conectividade, mas apresenta desvantagens.

No encaminhamento baseado em restrições, cada ligação tem um conjunto de propriedades associado e o percurso a determinar tem associado um conjunto restrições expressas em função dessas propriedades. O objectivo continua a ser determinar o caminho que minimize o custo, mas não viole nenhuma das restrições impostas. As restrições podem ser de desempenho (largura de banda, prioridade, atraso máximo) ou administrativas (o administrador pode impor um caminho).

Para o uso de EBR é necessário que o nó de origem tenha a capacidade de calcular o percurso tendo em conta os diferentes custos e as várias restrições, o que exige que o nó origem tenha acesso à informação sobre as restrições dos caminhos a determinar (informação local), a topologia da rede e propriedades associadas a cada ligação (informação a aobter dos outros nós da rede). Os percursos determinados recorrendo a EBR são denominados percursos explícitos, uma vez que são estabelecidos por outros meios que não o encaminhamento IP convencional.

Para estabelecer os percursos calculados, é necessário um protocolo de expedição de pacotes que suporte encaminhamento explícito.

2.4.1. Encaminhamento com restrições e MPLS

Nas redes IP tradicionais, o encaminhamento de pacotes é feito com base no endereço de destino sendo, portanto, impossível definir rotas diferentes calculadas pelos protocolos de encaminhamento convencionais.

Dadas as suas características, o protocolo MPLS é um excelente candidato a efectuar a expedição dos pacotes: existe uma separação entre os planos de expedição e controlo e a classificação dos pacotes e a consequente atribuição de um LSP é feita no LSR de fronteira, o que permite encaminhar explicitamente fluxos de tráfego evitando a introdução de mecanismos de classificação no núcleo da rede. Existem dois protocolos que permitem o estabelecimento de percursos explícitos para as LSPs e efectuar a reserva de recursos ao longo do percurso: o protocolo *Contraint Based Label Distribuition Protocol* (CR-LDP) e o RSVP com extensões para o estabelecimento de LSPs.

O protocolo CR-LDP, elaborado com base no protocolo LDP, introduz um conjunto de mecanismos adicionais que permitem o estabelecimento de LSPs com diversas restrições: restrições de encaminhamento explícito, QoS ou outras. Um LSP estabelecido com base em restrições denomina-se CR-LSP. Existem diversos objectos característicos deste protocolo: o ER (Explicit Route) é um campo das mensagens CR-LDP que especifica o caminho que um LSP deve tomar no momento em que está a ser estabelecido. É composto por um ou mais ER-Hops que constituem a especificação dos routers que fazem parte do caminho definido para o LSP [**2**].

O protocolo RSVP foi modificado por forma a suportar o estabelecimento de LSPs explicitas, com ou sem reserva de recursos, reencaminhamento de LSPs, preempção e detecção de ciclos. Para permitir o estabelecimento de percursos explícitos foi introduzido o objecto *Explicit Route Object* (ERO), cuja função e comportamento é semelhante à do objecto ER no CR-LDP.

2.5. Sobrevivencialidade em Redes MPLS

Por sobrevivencialidade de uma rede entende-se a sua capacidade de manter ininterruptos os serviços suportados quando existem falhas dos seus elementos, quer sejam ligações quer sejam nós.

Nas redes IP tradicionais, a falha de um elemento da rede implica que as tabelas de encaminhamento dos routers tenham que voltar a ser preenchidas, o que, dependendo do tipo de falha e das dimensões da rede, pode demorar segundos ou minutos.

Usando tecnologia MPLS, é possível reagir a falhas na rede de forma quase instantânea: podem ser utilizados LSPs de reserva que entram em funcionamento no caso de ocorrer uma falha que impeça o normal funcionamento dos LSPs principais. A protecção de falhas está dividida em quatro tipos: protecção de ligação, nó, caminho e segmento.

Na protecção de ligação pretende-se proteger um LSP de uma falha numa ligação de rede, pelo que o LSP de reserva é disjunto em ligações do LSP operacional nas ligações a proteger. Em caso de falhas, o tráfego é comutado para o LSP de reserva num dos nós extremos do LSP que falhou.

Com protecção de nó, o objectivo é proteger um LSP de falhas num nó, pelo que os LSPs operacional e de reserva são disjuntos no nó que requer protecção e consequentemente nas ligações desse nó. O tráfego é comutado para o LSP de protecção no nó imediatamente anterior ao que falhou.

A protecção de caminho visa salvaguardar a falha de qualquer elemento de rede, tendo os LSPs de reserva e operacional, percursos totalmente disjuntos, em nós e em ligações. A comutação para o LSP de reserva é feita nos extremos do LSP operacional.

Na protecção de segmento, a rede é dividida em vários domínios, sendo uma falha num domínio reparada dentro do próprio domínio.

Os LSPs de protecção podem ser estabelecidos antes ou depois da falha e os recursos necessários podem estar reservados *a priori* ou serem atribuídos a pedido depois da notificação de falha.

Os LSPs pré-estabelecidos com recursos reservados *a priori* garantem que, em caso de falha na rede, os compromissos de QoS sejam cumpridos. No caso de a reserva de recursos ser feita após a notificação da falha não é possível dar quaisquer garantias quanto à QoS então prestada. No entanto, a reserva de recursos *a priori* implica a não utilização de todos os recursos da rede e não permite estabelecer os LSPs de reserva pelo melhor caminho (no instante da falha).

2.6. Aplicações do MPLS

As principais aplicações do MPLS são nas áreas da Engenharia de Tráfego, fornecimento de Classes de Serviço e de Redes Privadas Virtuais (VPNs).

2.6.1. Engenharia de Tráfego

Engenharia de Tráfego é o processo de controlar a forma como o tráfego flui através de uma rede por forma a optimizar o seu desempenho e utilização de recursos evitando assim congestionamentos causados por uma utilização desigual da rede. Outro objectivo da Engenharia de Tráfego é possibilitar a operacionalização de redes fiáveis, incorporando

mecanismos que melhorem a integridade da rede, por forma a minimizar o impacto de erros e falhas da infra-estrutura nos serviços suportados.

A Engenharia de Tráfego é necessária na Internet principalmente porque os protocolos de encaminhamento mais comuns utilizam sempre os caminhos mais curtos para a expedição de pacotes. Este método de encaminhamento assegura a conectividade dentro da rede mas, uma vez que não existem mecanismos que permitam aos administradores da rede controlar eficazmente o percurso do tráfego e fazer partilha de carga entre dois caminhos com custos diferentes, são frequentes os seguintes problemas:

(i) os caminhos mais curtos entre diferentes pares de nós origem/destino sobrepõem-se em algumas ligações, o que provoca congestionamento nestas ligações;

(ii) o caminho mais curto entre dois nós encontra-se congestionado enquanto outros caminhos mais longos entre os mesmos dois nós estão subutilizados.

O MPLS é uma ferramenta bastante útil para a implementação de Engenharia de Tráfego, uma vez que:

- fornece rotas explícitas, o que permite aos *Label Switched Routers* (LSRs) de ingresso controlar com precisão a trajectória dos fluxos de tráfego;

- depois de estabelecidos os *Label Switched Paths* (LSPs), o caminho de um pacote dentro da rede MPLS é determinado pelo rótulo atribuído pelo LSR de ingresso;

- os administradores da rede podem atribuir propriedades às ligações e restrições aos LSPs;

- a distribuição de carga pode ser feita através de múltiplos LSPs entre o mesmo par origem/destino, mesmo que estes caminhos tenham diferentes custos;

- o uso de LSPs permite o cálculo mais simples e preciso de estatísticas de tráfego entre origens e destinos;

- podem ser usados LSPs de reserva para evitar a degradação dos serviços suportados em caso de falha dos equipamentos de rede.

2.6.2. Redes Privadas Virtuais (VPN)

Uma Rede Privada Virtual (VPN) simula a operação de uma *Wide Area Network* (WAN) sobre uma rede pública. Para poder prestar aos seus clientes um serviço VPN, o operador

de rede tem que garantir a confidencialidade dos dados e resolver o problema da utilização de endereços IP privados, não únicos, na sua rede.

Através do MPLS, estas questões são resolvidas de forma simples e eficiente, criando LSPs que unem os diferentes sites de cada VPN, como ilustrado na Figura 2 – Redes Privadas Virtuais (VPN) usando MPLS.

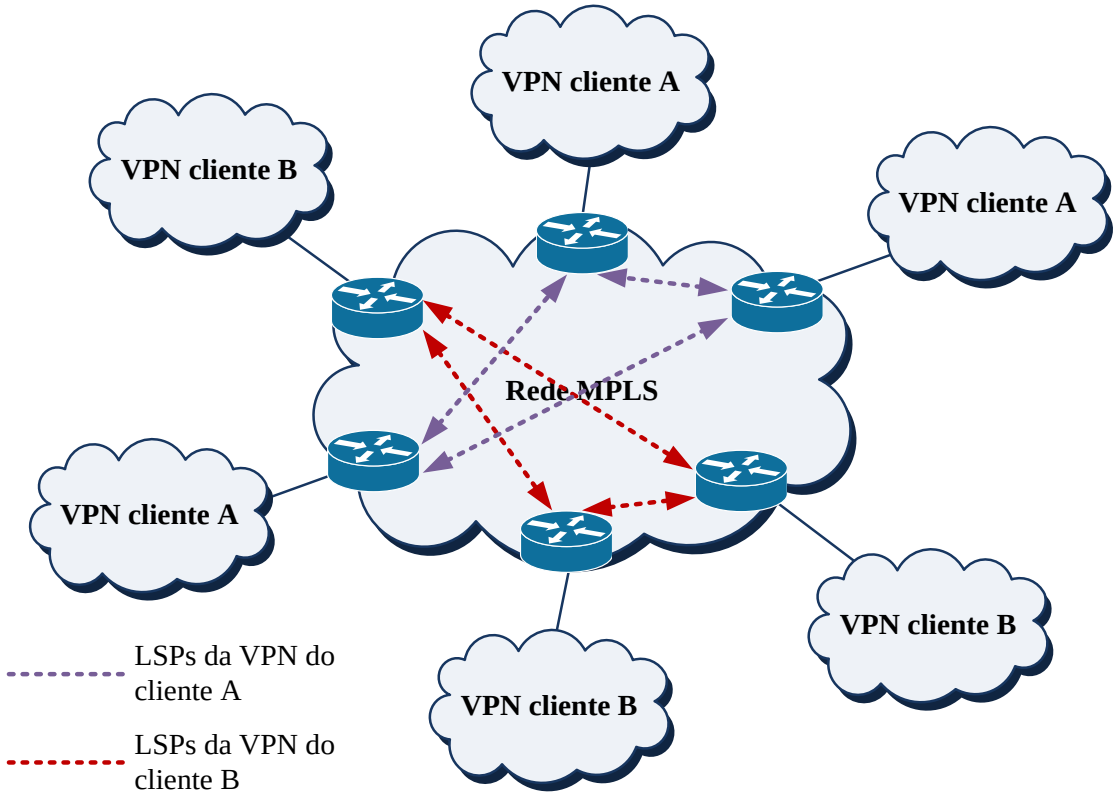


Figura 2 – Redes Privadas Virtuais (VPN) usando MPLS

O tráfego é classificado, à entrada da rede MPLS, com base no endereço de destino VPN a que pertence. Desta classificação resulta a atribuição de um rótulo e, consequentemente, de um LSP. Desta forma, o tráfego entre as duas VPNs é separado de forma lógica.

2.6.3. Classes de Serviço

O MPLS pode servir de suporte ao modelo *Differentiated Services (DiffServ)*. Este modelo especifica um conjunto de mecanismos que permite classificar o tráfego num número limitado de classes de serviço, que merecem por parte da rede um tratamento distinto. Esta

classificação baseia-se no campo *Type of Service* (ToS) existente no cabeçalho dos pacotes IPv4.

As classes de serviço permitem disponibilizar ao cliente vários serviços, desde os *best effort* (para transferência de ficheiros, por exemplo) até serviços sensíveis ao atraso, como voz e vídeo. No entanto, esta classificação por si só não permite QoS, sendo necessário empregar mecanismos de Engenharia de Tráfego.

O MPLS pode ser utilizado em conjunto com o modelo *DiffServ* de duas formas: utilizar um LSP entre cada par Origem/Destino e utilizar o campo CoS para diferenciar o tratamento dado aos pacotes ou utilizar um LSP entre cada par Origem/Destino para cada classe de serviço.

2.7. MPLS-TE – Multiprotocol Label Switching with Traffic Engineering support

O encaminhamento convencional baseado em algoritmos IGP (*Interior Gateway Protocol*) não proporciona uma distribuição do tráfego de forma balanceada, ou seja, alguns recursos podem ser subutilizados enquanto outros podem sofrer com cargas excessivas de tráfego.

Um indicador limitado sobre Engenharia de Tráfego pode ser fornecido manipulando as métricas do IGP associadas às diferentes ligações da rede. No entanto, estas informações são difíceis de administrar em ambientes com várias opções de encaminhamento entre dois pontos: as métricas do estado da ligação que são determinadas pelos protocolos de encaminhamento e a forma como são manipuladas no domínio MPLS são relativamente diferentes das que são usadas nos serviços integrados (arquitectura IntServ) de uma rede IP tradicional. Uma vez calculado o caminho usando as métricas do IGP, torna-se necessária sinalização para o implementar.

Duas possíveis soluções foram desenvolvidas pelo *IETF MPLS Working Group*: o CR_LDP e o RSVP-TE. É interressante observar que o *IETF MPLS Working Group* decidiu abandonar os trabalhos focados no desenvolvimentos do CR-LDP a fim de concentrar esforços no protocolo RSVP-TE como o protocolo de sinalização para aplicações de Engenharia de Tráfego com MPLS.

2.7.1. CR-LDP – Constraint Based Routed – Label Distribution Protocol

O CR-LDP é construído sobre o LDP, que já é parte do MPLS. Embora os estudos do *IEFT MPLS Working Group* tenham sido abandonados, este protocolo possui resultados satisfatórios e não implica a implementação de um novo protocolo, com o consequente aumento na carga de processamento, tal como acontece com o preferido RSVP-TE.

Assim como o LDP, o CR-LDP usa um esquema de codificação denominado TLV (*Type-Lenght-Value*). Tratam-se de mensagens passadas através da rede, que estão divididas em três campos básicos: o campo *type*, que define o tipo de mensagem; o campo *length*, que especifica o comprimento do campo seguinte (*value*) em bytes; o campo *value*, de tamanho *length*, codifica a informação que está interpretada de acordo com o *type*. A manipulação destes campos permite implementar Engenharia de Tráfego no domínio MPLS.

O CR-LDP suporta encaminhamento explícito do tipo *strict*, em que o caminho completo a ser seguido é fixo, e encaminhamento explícito do tipo *loose*, onde somente alguns routers são nós fixos de um caminho. Utiliza o protocolo UDP (*User Datagram Protocol*) para descobrir novos pontos num domínio MPLS e o TCP (*Transfer Control Protocol*) para realização de controlo, gestão, *label request* e *label mapping*.

A sinalização usando o CR-LDP é ilustrada na Figura 3 e descrita resumidamente nos próximos parágrafos.

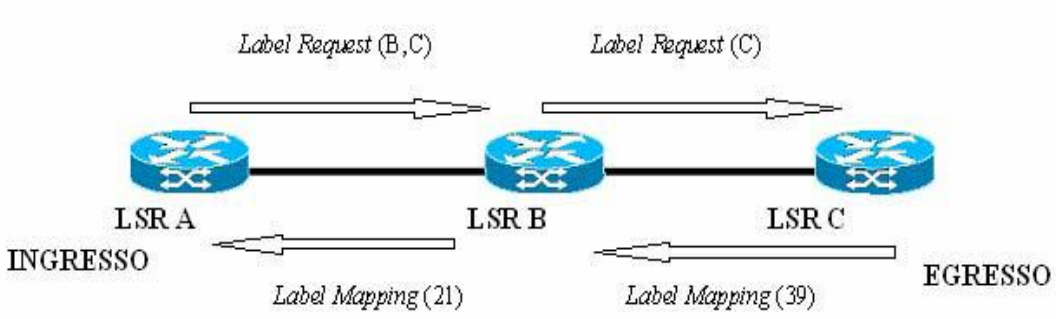


Figura 3 – Sinalização CR-LDP [3]

O LSR A (de ingresso) determina que é necessário criar um novo LSP para o LSR C. Os parâmetros de tráfego requeridos para o encaminhamento ou as políticas administrativas para a rede indicam ao LSR A para determinar um novo LSP através do LSR B, lembrando que o número de *hops* destino não é factor determinante para a sua determinação.

O LSR A constrói uma mensagem de *label request* para a determinação da rota explícita, indicando detalhes dos parâmetros de tráfego requeridos para a nova rota. Então, o LSR A reserva os recursos requeridos para este novo LSP e encaminha para o LSR B a mensagem de *label request* numa sessão TCP.

O LSR B recebe esta mensagem e percebe que não é o equipamento de saída para este LSP. Então, caso seja possível, o LSR B reserva os recursos pedidos para o novo LSP, modifica a informação para o encaminhamento explícito na mensagem de *label request* e encaminha-a para o LSR C.

O LSR C percebe que é o equipamento de saída para o novo LSP e realiza as reservas de recursos requeridas, caso seja possível. Reserva então um rótulo para o LSR B através de uma mensagem de *label mapping*, que contém detalhes finais sobre os parâmetros reservados para o LSP. O LSR B recebe a correspondente mensagem e passa para o LSR A um novo rótulo através me uma mensagem de *label mapping*.

Finalmente, no LSR A ocorre um processamento semelhante para a criação deste LSP, com excepção do o envio de mensagens para designação de rótulos, já que este se trata do LSR de ingresso para este LSP [**3**].

2.7.2. RSVP-TE – Resource Reservation Protocol – Traffic Engineering

Criado inicialmente pelo IETF para aplicações em serviços integrados (*Intserv*), o RSVP foi desenvolvido para ser um mecanismo de sinalização com o objectivo de reservar recursos através de uma rede, permitindo a um *host* especificar uma determinada requisição de serviço para um certo fluxo na rede.

As extensões do RSVP para Engenharia de Tráfego proporcionaram uma excelente adequação deste protocolo para a distribuição de rótulos MPLS de forma optimizada. Desde Fevereiro de 2003 que o *IETF MPLS Working Group* passou a concentrar os seus esforços no RSVP-TE para sinalização e estabelecimento de rotas num ambiente MPLS.

O processo de sinalização usando o RSVP-TE é bastante parecido com o do CR-LDP, tal como ilustra a Figura 4.

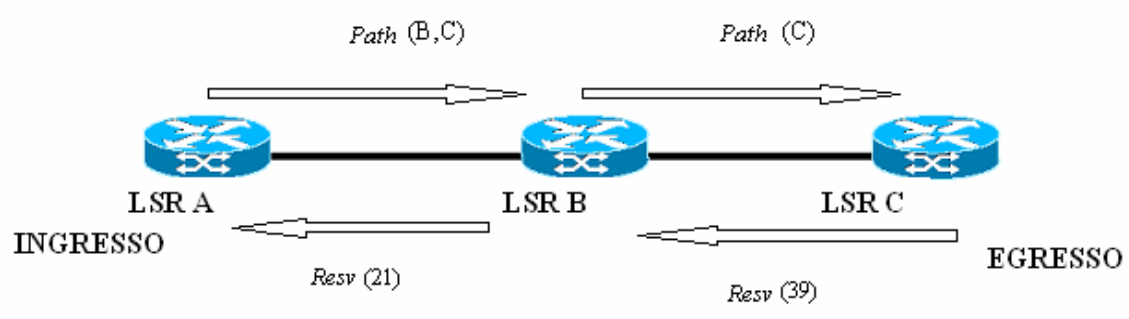


Figura 4 – Sinalização RSVP-TE [3]

No processo de sinalização utilizando o RSVP-TE o LSR A (de ingresso) tem a função de criar um LSP até ao LSR C. Os parâmetros de tráfego requeridos para o encaminhamento ou as políticas administrativas para a rede indicam ao LSR A para calcular um novo LSP através do LSR B. É importante ressalvar que a menor distância não é factor determinante para a escolha desta rota. O LSR A cria uma mensagem *Path* para determinação da rota explícita e também com detalhes dos parâmetros de tráfego requeridos para a nova rota. Então, o LSR A reserva os recursos requeridos para este novo LSP e encaminha para o LSR B a menagem de *Path* num datagrama IP.

O LSR B recebe esta mensagem e percebe que não é o equipamento de saída para este LSP. Então, caso possível, o LSR B reserva os recursos pedidos para o novo LSP, modifica a informação para o encaminhamento explícito na mensagem *Path* e encaminha para o LSR C. O LSR C percebe que é o equipamento de saída para o novo LSP e realiza as reservas de recursos requeridas, caso seja possível. Aloca então um rótulo para este novo LSP e distribui o rótulo para o LSR B através de uma mensagem *Resv*, que contém detalhes finais sobre os parâmetros reservados para o LSP. O LSR B recebe a mensagem, finaliza a reserva, actualiza as tabelas correspondentes e passa para o LSR A um novo rótulo através de uma mensagem *Resv*, e tal como quando utilizamos o CR-LDP, o LSR A reserva os recursos necessários e não aloca ou encaminha para um LSR ascendente (*upstream*) um novo rótulo já que ele é o LSR de ingresso, finalizando assim o estabelecimento do LSP [**3**].

2.7.3. Comparação entre o CR-LDP e o RSVP-TE

Relativamente ao conteúdo das mensagens, as semelhanças entre os protocolos CR-LDP e RSVP-TE são inúmeras. Basicamente o conteúdo é o mesmo, excepto em alguns pequenos detalhes. Ambos transportam as mesmas informações para o estabelecimento das rotas.

A principal diferença no funcionamento dos protocolos prende-se com as implicações decorrentes do facto do RSVP-TE ser do tipo *soft-state*, enquanto o CR-LDP é do tipo *hard-state*. O facto de o RSVP-TE ser *soft-state* proporciona um *overhead* adicional, o que requer que mensagens de *refresh* sejam enviadas periodicamente entre cada nó da rede para manutenção de um LSP.

Algumas alterações foram propostas para solucionar esse problema no RSVP-TE e minimizar os efeitos do *soft-state*, como a introdução de um mecanismo de reconhecimento de mensagem recebida (*acknowledge*), tornando o protocolo de troca de mensagens confiável, possibilitando reduzir o tempo de *refresh* dos estados e consequentemente o *overhead*. Outro mecanismo para redução do *overhead* é a possibilidade de com apenas uma mensagem realizar o *refresh* de vários estados simultaneamente.

Outra diferença essencial no funcionamento destes protocolos é o mecanismo de rápida notificação de falhas presentes no RSVP-TE, implementado pela mensagem *Notification*. O protocolo CR-LDP também possui uma mensagem de notificação (*Notification*), no entanto as funções de ambas as mensagens são distintas. No CR-LDP quando é detectada uma falha numa rota ela é propagada com as mensagens *Release/Winthdraw* a partir do ponto de falha. Os recursos alocados devem ser libertados nesta fase. A mensagem de notificação serve para informar falhas no processamento de mensagens.

Já no RSVP-TE a mensagem de notificação informa sobre falhas em rotas e é enviada directamente do ponto de detecção ao ponto de reparação (um LSR responsável por realizar a restauração do LSP).

Na Tabela 1 são identificadas algumas semelhanças entre ambos os protocolos, enquanto que a Tabela 2 mostra as suas principais diferenças.

Características	CR-LDP	RSVP-TE
Sinalização Inicial	Mensagens de *label request*	Mensagens de *Path* contendo o pedido do rótulo
Sinalização de Confirmação	Mensagens de *label mapping*	Mensagens RESV
Definição para Serviços Diferenciados (*DiffServ*)	DIFFSERV_PSC TLV	DIFFSERV_PSC *object*
Suporte para LSPs *point-to-multipoint*	Não	Não
Capacidade para rotas explícitas	Definida pela sinalização contida no TLV	Carregada no objecto EXPLICIT_ROUTE

Tabela 1 – Semelhanças entre o CR-LDP e o RSVP-TE

Características	CR-LDP	RSVP-TE
Estágio de Desenvolvimento	Recente	Antigo, com novas extensões adicionadas para Engenharia de Tráfego
Transporte de sinalização	UDP para novas descobertas, TCP para manutenção da sessão	Datagramas IP ou encapsulamento UDP
Estado da ligação	*Hard state*	*Soft-state*
Confiabilidade	Falhas produzem uma resposta da sinalização	Depende do tempo entre as mensagens de *refresh*
Extensibilidade	TLVs experimentais	Objectos experimentais
Escalabilidade	Ligações *hard state* reduzem o processamento de sinalização	Requer redução de mensagens de *refresh*, agregação para diminuir o processamento.
Interoperabilidade	Bem definida, suporta a maioria dos protocolos da camada de ligação de dados	O estabelecimento de túneis através de redes ATM pode ser configurado manualmente

Tabela 2 – Diferenças entre o CR-LDP e o RSVP-TE

Em suma, o MPLS aparece no cenário de redes como uma arquitectura emergente, baseada num modelo de encaminhamento de pacotes, sendo a tecnologia que se apresenta mais promissora na tentativa de melhorar o desempenho das redes, por ser flexível e por permitir o mapeamento em várias tecnologias de rede. As suas melhores perspectivas residem na possibilidade de adicionar à tecnologia IP o paradigma de circuito virtual e a

possibilidade de aplicar conceitos como a Engenharia de Tráfego e a garantia de QoS sem a necessidade de alterar totalmente a estrutura já existente nas redes de dados actuais.

A Engenharia de Tráfego torna-se cada vez mais necessária, não somente pelo seu carácter técnico mas também como factor económico, já que um uso mais inteligente da rede cria a possibilidade de não ser necessária a actualização dos equipamentos para atender a alguma carência como, por exemplo, largura de banda de transmissão.

Ambos os protocolos de sinalização analisados, o CR-LDP e o RSVP-TE, disponibilizam funcionalidades semelhantes para a implementação de Engenharia de Tráfego. Não há um motivo determinante para a escolha do RSVP-TE, mas o facto deste protocolo ser largamente utilizado na reserva de recursos das redes IP tradicionais é um aspecto determinante, havendo apenas a necessidade de adicionar a capacidade de distribuição de rótulos, o que é feito através das extensões que foram propostas ao protocolo RSVP.

2.8. MPLS TE

Traffic Engineering (TE) é o processo de condução do tráfego através do *backbone* da rede de forma a facilitar o uso eficiente da largura de banda entre cada par de routers. Anteriormente, esta operação era realizada recorrendo a IP ou a ATM, dependendo do protocolo que estivesse a ser utilizado.

A principal vantagem da implementação do MPLS-TE assenta no facto deste protocolo proporcionar a combinação das capacidades do ATM (*Asynchronous Transfer Mode*) TE com a diferenciação de classes de serviço proporcionada pelo IP. O MPLS TE permite a construção de LSPs (*Label Switched Paths*) através dos quais é feito o envio da informação. Os LSPs do MPLS-TE deixam o *Headend* do túnel TE controlar o caminho que o seu tráfego toma para um determinado destino. Este método é mais flexível do que encaminhar o tráfego com base apenas no endereço de destino.

O MPLS-TE evita problemas de *flooding* que o ATM e outros modelos *overlay* apresentam. Em vez de formar adjacências automáticas entre LSPs-TE, o MPLS-TE usa um mecanismo denominado por *autoroute* para construir uma tabela de encaminhamento utilizando LSPs MPLS-TE sem formar uma rede completa de vizinhos. Tal como o ATM, o MPLS-TE reserva largura de banda na rede quando cria LSPs. Ao reservar largura de

banda para um LSP é introduzido o conceito de recurso consumível na rede. Se são criados LSPs-TE que reservam largura de banda enquanto são adicionadas LSPs à rede, é possível encontrar caminhos alternativos através da rede que possuam largura de banda disponível para serem reservados. Ao contrário do ATM, não existe uma obrigação de cumprimento dos parâmetros da reserva efectuados. A reserva é apenas realizada com fundamentos de controlo, ou seja, se um LSR realiza uma reserva de 10Mb e envia 100Mb através do LSP a rede tenta entregar os 100Mb, excepto se forem introduzidas políticas de QoS na origem dos dados.

2.9. Redes Privadas Virtuais (VPN)

Uma Rede Privada Virtual (VPN) é definida na maioria das vezes como uma rede em que cada cliente tem uma ligação a vários sites de uma forma distribuída, através duma infra-estrutura que partilha as mesmas políticas de acesso e segurança de uma rede privada. Os recentes acontecimentos de marketing que rodeiam o tema VPNs, desde as novas tecnologias que suportam VPNs até a uma multiplicidade de produtos e serviços relacionados com VPNs, podem levar-nos a pensar que o conceito de VPN é dos maiores e mais rentáveis conceitos tecnológicos. Contudo, tal como acontece na maioria dos casos, o conceito de VPN é uma conceito que tem mais de 10 anos e é bastante bem conhecido pelos operadores de serviços no mercado. As novas tecnologias e produtos simplesmente permitem mais fiabilidade, escalabilidade e são mais eficazes a nível de custo, para uma implementação dos mesmos produtos. Com a redução de custos e uma melhor escalabilidade, não é surpresa que os serviços VPN sejam um dos maiores propulsores do uso da tecnologia MPLS.

A idéia de utilizar uma rede pública como a Internet em vez de linhas privativas para implementar redes corporativas é denominada de *Virtual Private Network* (VPN) ou Rede Privada Virtual. As VPNs são túneis de criptografia entre pontos autorizados, criados através da Internet ou outras redes públicas e/ou privadas para transferência de informações, de modo seguro, entre redes corporativas ou utilizadores remotos.

A segurança é a primeira e mais importante função da VPN. Uma vez que dados privados serão transmitidos pela Internet, que é um meio de transmissão inseguro, eles devem ser protegidos de forma a não permitir que sejam modificados ou interceptados.

Outro serviço oferecido pelas VPNs é a ligação entre empresas (Extranets) através da Internet, além de possibilitar ligações *dial-up* criptografadas que podem ser muito úteis para utilizadores móveis ou remotos, bem como para filiais distantes de uma empresa.

Uma das grandes vantagens decorrentes do uso das VPNs é a redução de custos das comunicações corporativas, pois elimina a necessidade de *links* dedicados de longa distância que podem agora ser substituídos pela Internet. As LANs podem, através de *links* dedicados ou do tipo dial-up, ligar-se a algum fornecedor de acesso local e interligar-se a outras LANs, possibilitando o fluxo de dados através da Internet. Esta solução pode ser bastante interessante sob o ponto de vista económico, sobretudo nos casos em que estão envolvidas ligações internacionais ou nacionais de longa distância. Outro factor que simplifica a operacionalização da WAN é que a ligação LAN-Internet-LAN fica parcialmente a cargo dos fornecedores de acesso.

2.9.1. Evolução

Inicialmente as redes de computadores foram implementadas sobre duas tecnologias: linhas alugadas para conectividade permanente e linhas de *dial-up* para uso ocasional do serviço. A figura seguinte (Figura 5) apresenta uma rede típica dessa época [**4**].

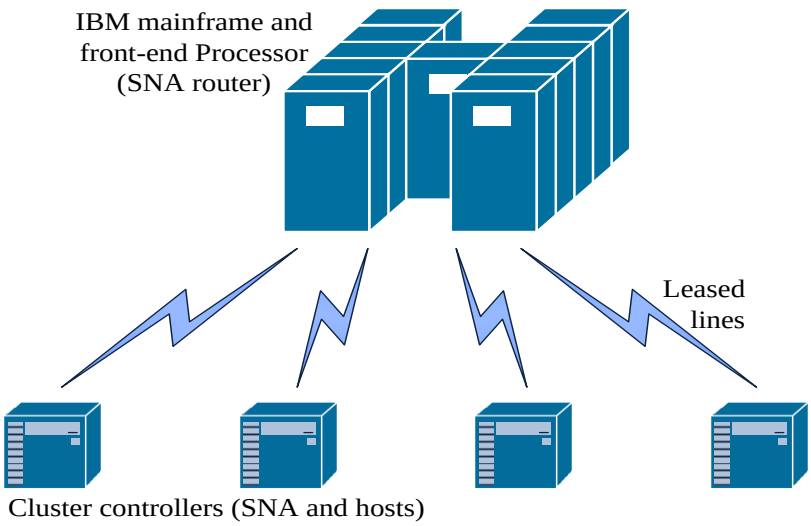


Figura 5 – Rede de computadores típica de há 15 anos atrás (Adaptada de [4])

A rede de computadores inicialmente implementada fornecia aos clientes boa segurança, contudo não implementava uma relação muito boa entre o serviço e o custo efectivo devido principalmente a duas razões:

- o perfil de tráfefo típico numa rede entre dois quaisquer sites varia consoante a hora do dia, o dia do mês e até mesmo com as épocas do ano;

- o consumidor final está interessado em respostas rápidas, resultando numa largura de banda elevada entre sites, mas a largura de banda disponível nas linhas alugadas é apenas utilizada uma parte do tempo (quando os respectivos utilizadores estão ligados).

Estas duas razões levaram a indústria de comunicações de dados e de serviços a desenvolver e implementar um conjunto de esquemas de multiplexagem estatística que fornecia aos clientes um serviço equivalente ao do aluguer de linhas. Este serviço era contudo mais barato, devido aos benefícios resultantes da multiplexagem estatística, uma vez que o operador podia atingir um número muito maior de clientes. A primeira rede privada virtual era baseada em tecnologias como *X.25* e *Frame Relay*, e mais tarde *SMDS* e *ATM*. A Figura 6 permite ilustrar uma rede VPN típica baseada em tecnologia *Frame Relay*.

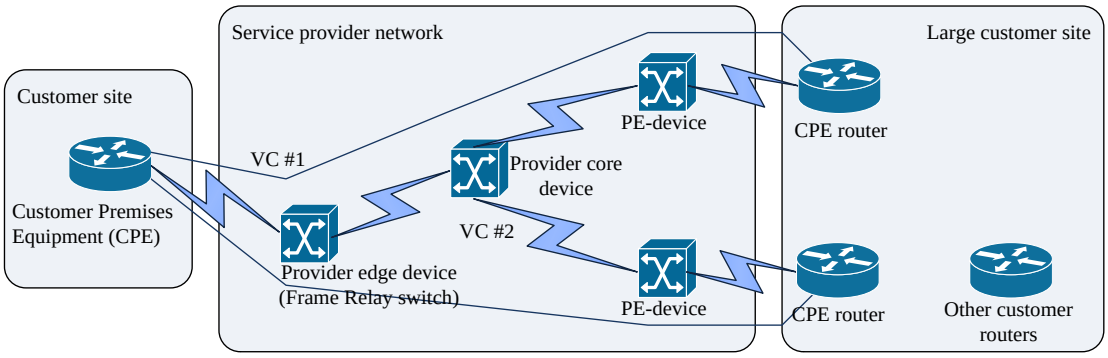


Figura 6 – Rede *Frame Relay* típica (Adaptada de [4])

2.9.2. VPNs modernas

Com a introdução de novas tecnologias nos prestadores de serviços de redes e com as novas exigências dos clientes, o conceito VPN tornou-se cada vez mais complexo. Os vendedores introduziram novos termos e muitas vezes alguns deles bastante conflituosos, o que ainda veio aumentar mais a complexidade.

Os modernos serviços VPN podem abranger uma grande variedade de tecnologias e topologias. A única maneira de lidar com esta diversidade foi introduzir classificações VPN, que podem ser baseadas nos quatro critérios seguintes:

- o problema das empresas que a tecnologia tenta resolver: os problemas associados ás diversas classes de negócio são as comunicações intra-companhia (denominadas por *intranet*), inter-companhias (também denominadas de *extranet*) e acesso para utilizadores móveis (denominados *Virtual Private Dialup Network*).

- a camada OSI onde o fornecedor de serviços troca a informação de topologia com o cliente: as categorias principais são o modelo *overlay*, onde o fornecedor de serviços proporciona ao cliente um único conjunto de ligações ponto-a-ponto entre sites de cliente, e o modelo *peer*, onde o fornecedor de serviço e o cliente trocam informação de *routing* da camada 3;

- a tecnologia sobre a qual assenta a camada 2 ou a camada 3 é usada para implementar os serviços VPN dentro do fornecedor de serviços de rede, podendo ser *X.25*, *Frame Relay*, *SMDS*, *ATM* ou *IP*.

- a topologia de rede, que pode variar desde a topologia *hub-and-spoke* até ás topologias em malha e multi-nível hierárquico, para redes de grande escala.

2.9.3. Aplicações

Seguidamente são apresentadas as três aplicações mais importantes das VPNs.

Acesso remoto via Internet

O acesso remoto a redes corporativas através da Internet pode ser viabilizado com uma VPN através da ligação local a um fornecedor de acesso (*Internet Service Provider* - ISP). A estação remota liga ao fornecedor de acesso, ligando-se à Internet, e o software de VPN cria uma rede virtual privada entre o utilizador remoto e o servidor de VPN corporativo através da Internet.

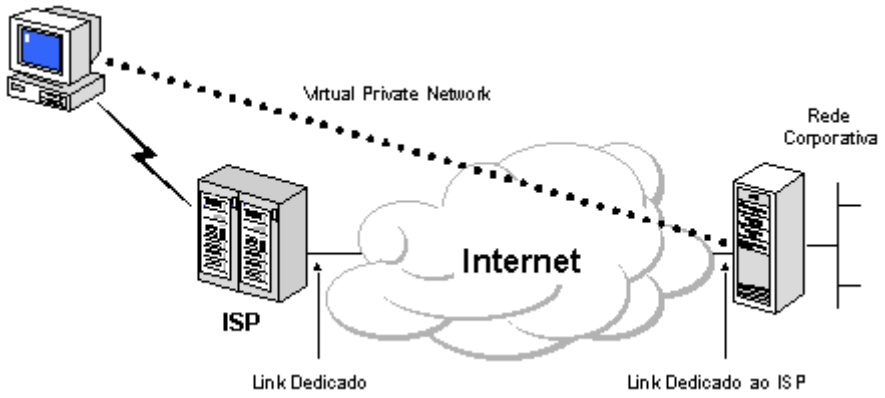


Figura 7 – Acesso remoto via Internet [5]

Ligação de LANs via Internet

Uma solução que substitui as ligações entre LANs recorrendo a circuitos dedicados de longa distância é a utilização de circuitos dedicados locais que interligam as LANs através da Internet. O software de VPN assegura esta interligação formando a WAN corporativa. Dependendo das aplicações, pode-se optar pela utilização de circuitos *dial-up* numa das pontas, devendo a LAN corporativa estar preferencialmente ligada à Internet através de circuito dedicado local, ficando disponível 24 horas por dia para tráfego proveniente da VPN.

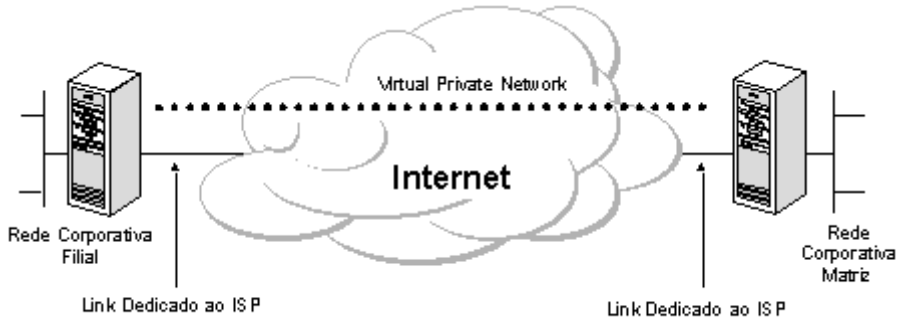


Figura 8 – Ligação de LANs via Internet [5]

Ligação de computadores numa Intranet

Em algumas organizações, existem dados confidenciais cujo acesso é restrito a um pequeno grupo de utilizadores. Nestas situações, redes departamentais locais são implementadas fisicamente de forma separada da LAN corporativa. Esta solução, apesar de

garantir a "confidencialidade" das informações, cria dificuldades de acesso a dados da rede corporativa por parte dos departamentos isolados.

As VPNs possibilitam a ligação física entre redes locais, restringindo acessos indesejados através da inserção de um servidor de VPNs entre elas. O servidor VPN não irá actuar como um router entre a rede departamental e o resto da rede corporativa, uma vez que o router possibilitaria a ligação entre as duas redes permitindo o acesso de qualquer utilizador à rede departamental sensível. Com o uso da VPN o administrador da rede pode definir quais os utilizadores que estão credenciados a atravessar o servidor VPN e a aceder aos recursos da rede departamental restrita. Adicionalmente, toda a comunicação ao longo da VPN pode ser criptografada, assegurando a "confidencialidade" das informações. Os restantes utilizadores não credenciados não irão sequer ver a rede departamental.

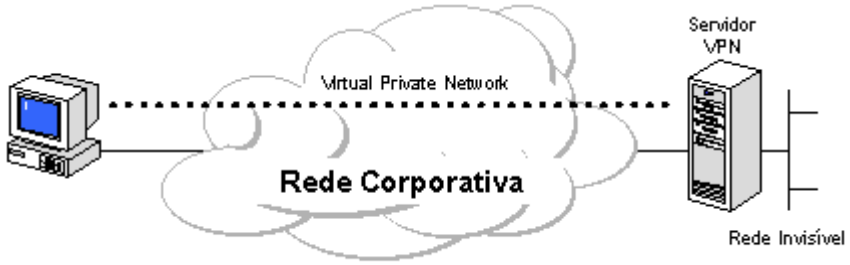


Figura 9 – Ligação de computadores numa Intranet [5]

2.9.4. Requisitos Básicos de uma rede VPN

No desenvolvimento de soluções de rede é bastante desejável que sejam implementadas facilidades de controle de acesso a informações e a recursos corporativos. A VPN deve dispor de recursos para permitir o acesso de clientes remotos autorizados aos recursos da LAN corporativa, viabilizar a interligação de LANs de forma a possibilitar o acesso de filiais, compartilhando recursos e informações e, finalmente, assegurar privacidade e integridade dos dados ao atravessar a Internet e a própria rede corporativa. A seguir são enumeradas as características mínimas desejáveis numa VPN:

- **Autenticação de Utilizadores:** verificação da identidade do utilizador, restringindo o acesso às pessoas autorizadas. Deve dispor de mecanismos de

auditoria, fornecendo informações referentes aos acessos efectuados - quem acedeu, a quê e quando.

- **Gestão de Endereços**: o endereço do cliente na sua rede privada não deve ser divulgado, devendo-se adoptar endereços fictícios para o tráfego externo.

- **Criptografia de Dados**: os dados devem atravessar a rede pública ou privada num formato cifrado e, caso sejam interceptados por utilizadores não autorizados, não deverão ser descodificados, garantindo a privacidade da informação. O reconhecimento do conteúdo das mensagens deve ser exclusivo dos utilizadores autorizados.

- **Gestão de Chaves**: o uso de chaves que garantem a segurança das mensagens criptografadas deve funcionar como um segredo compartilhado exclusivamente entre as partes envolvidas. A gestão de chaves deve garantir a troca periódica das mesmas, visando manter a comunicação de forma segura.

- **Suporte a Múltiplos Protocolos**: com a diversidade de protocolos existentes, torna-se bastante desejável que uma VPN suporte protocolos padrão de facto usados nas redes públicas, tais como IP (*Internet Protocol*) e o IPX (*Internetwork Packet Exchange*).

2.9.5. Túneis

As redes virtuais privadas baseiam-se na tecnologia de tunelamento, cuja existência é anterior às VPNs, e que pode ser definida como o processo de encapsular um protocolo dentro de outro. O uso do tunelamento nas VPNs incorpora um novo componente a esta técnica: antes de encapsular o pacote que será transportado, este é criptografado de forma a ficar ilegível caso seja interceptado durante o seu transporte. O pacote criptografado e encapsulado viaja através da Internet até alcançar o seu destino, onde é desencapsulado e decriptografado, retornando ao seu formato original. Uma característica importante é que pacotes de um determinado protocolo podem ser encapsulados em pacotes de protocolos diferentes. Por exemplo, pacotes de protocolo IPX podem ser encapsulados e transportados dentro de pacotes TCP/IP.

O protocolo de tunelamento encapsula o pacote com um cabeçalho adicional que contém informações de encaminhamento que permitem a travessia dos pacotes ao longo da rede intermédia. Os pacotes encapsulados são encaminhados na rede intermédia entre as extremidades do túnel. Túnel é a denominação do caminho lógico percorrido pelo pacote ao longo da rede intermédia. Após alcançar o seu destino, o pacote é desencapsulado e encaminhado para o seu destino final. A rede intermédia por onde o pacote passa pode ser qualquer rede pública ou privada. Note-se que o processo de tunelamento envolve encapsulamento, transmissão ao longo da rede intermédia e desencapsulamento do pacote.

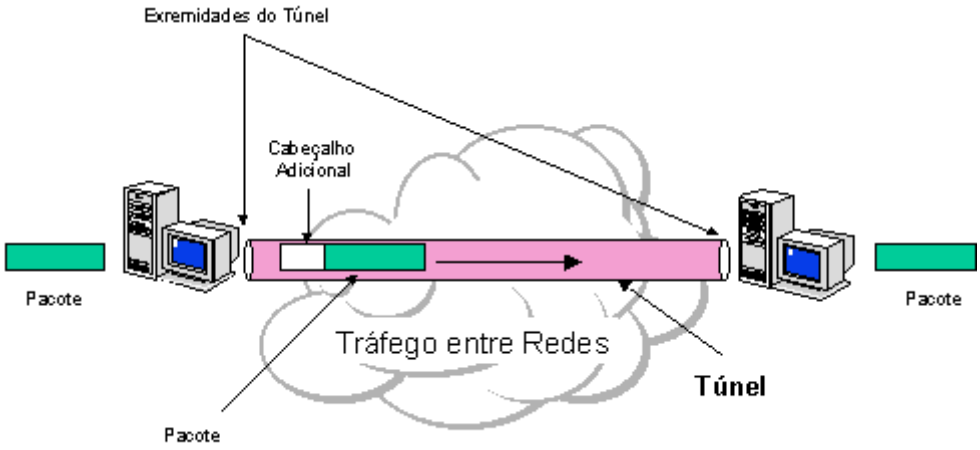


Figura 10 – Túneis [5]

2.9.6. Protocolos de Tunelamento

Para se estabelecer um túnel é necessário que as suas extremidades utilizem o mesmo protocolo de tunelamento. O tunelamento pode ocorrer nas camadas 2 ou 3 do modelo de referência OSI (*Open Systems Interconnection*).

- **Tunelamento de Nível 2 - (PPP sobre IP)**

O objetivo é transportar protocolos de nível 3, tais como o IP e IPX, na Internet. Os protocolos utilizam tramas, encapsulando os pacotes da camada 3 (como IP/IPX) em tramas PPP (*Point-to-Point Protocol*). Como exemplos de protocolos podemos referir [**5**]:

- PPTP (*Point-to-Point Tunneling Protocol*) da Microsoft, que permite que o tráfego IP, IPX e NetBEUI seja criptografado e encapsulado para ser enviado através de redes IP privadas ou públicas;
- L2TP (*Layer 2 Tunneling Protocol*) do IETF (*Internet Engineering Task Force*), que permite que o tráfego IP, IPX e NetBEUI seja criptografado e enviado através de canais de comunicação ponto a ponto, tais como IP, X25, Frame Relay ou ATM.
- L2F (*Layer 2 Forwarding*) da Cisco, utilizado para VPNs do tipo *dial-up*.

- **Tunelamento ao Nível 3 - (IP sobre IP)**

Encapsulam pacotes IP com um cabeçalho adicional IP antes de os enviar através da rede.

- *IP Security Tunnel Mode* (IPSec) do IETF, que permite que pacotes IP sejam criptografados e encapsulados com cabeçalho IP adicional para serem transportados numa rede IP pública ou privada. O IPSec é um protocolo desenvolvido para IPv6, devendo no futuro constituir-se como padrão para todas as formas de VPN, caso o IPv6 venha a substituir de facto o IPv4. Entretanto, o IPSec sofreu adaptações que possibilitam também a sua utilização com o IPv4.

2.9.7. O funcionamento dos Túneis

Nas tecnologias orientadas à camada 2, um túnel é semelhante a uma sessão onde as duas extremidades do túnel negoceiam a configuração dos parâmetros para o estabelecimento do túnel, tais como endereçamento, criptografia e parâmetros de compressão. Na maior parte das vezes, são utilizados protocolos que implementam o serviço de datagrama. A gestão do túnel é realizada através de protocolos de manutenção. Nestes casos, é necessário que o túnel seja criado, mantido e fechado. Nas tecnologias de camada 3, não existe a fase de manutenção do túnel.

Assim que o túnel é estabelecido os dados podem ser enviados. O cliente ou servidor do túnel utiliza um protocolo de tunelamento na transferência de dados que adiciona um cabeçalho, preparando o pacote para o transporte. Só então o cliente envia o pacote encapsulado para a rede que o encaminhará até ao servidor do túnel. Este recebe o

pacote, desencapsula-o removendo o cabeçalho adicional e encaminha o pacote original à rede de destino. O funcionamento entre o servidor e o cliente do túnel é semelhante.

2.9.8. Protocolos vs Requisitos de Tunelamento

Os protocolos de nível 2, tais como PPTP e L2TP, foram baseados no PPP e, como consequência, herdaram muito das suas características e funcionalidades. Estas características e as suas correspondentes no nível 3 são analisadas conjuntamente com alguns dos requisitos básicos das VPNs.

Autenticação de utilizador

Os protocolos de tunelamento da camada 2 herdaram os esquemas de autenticação do PPP e os métodos EAP (*Extensible Authentication Protocol*). Muitos esquemas de tunelamento da camada 3 assumem que as extremidades do túnel são conhecidas e autenticadas antes mesmo que ele seja estabelecido. Uma excepção é o IPSec que fornece a autenticação mútua entre as extremidades do túnel. Na maioria das implementações deste protocolo, a verificação dá-se a nível de máquina e não do utilizador. Como resultado, qualquer utilizador com acesso às máquinas que funcionam como extremidades do túnel pode utilizá-lo. Esta falha de segurança pode ser suprida quando o IPSec é usado em conjunto com um protocolo da camada de ligação de dados, como o L2TP.

Suporte de *token card*

Com a utilização do EAP, os protocolos de tunelamento da camada de ligação dados podem suportar uma variedade de métodos de autenticação, tais como senhas e cartões inteligentes (*smart cards*). Os protocolos da camada 3 também podem usar métodos similares: por exemplo, o IPSec define a autenticação de chave pública durante a negociação de parâmetros feita pelo ISAKMP (*Internet Security Association and Key Management Protocol*).

Endereçamento dinâmico

O tunelamento na camada 2 suporta alocação dinâmica de endereços baseada nos mecanismos de negociação do NCP (*Network Control Protocol*). Normalmente, esquemas de tunelamento na camada 3 assumem que os endereços foram atribuídos antes da inicialização do túnel.

Compressão de dados

Os protocolos de tunelamento da camada 2 suportam esquemas de compressão baseados no PPP. O IETF está a propor mecanismos semelhantes, tais como a compressão de IP, para o tunelamento na camada 3.

Criptografia de dados

Protocolos de tunelamento na camada de ligação de dados suportam mecanismos de criptografia baseados no PPP. Os protocolos de nível 3 também podem usar métodos similares. No caso do IPSec, são definidos vários métodos de criptografia de dados que são executados durante o ISAKMP. Algumas implementações do protocolo L2TP utilizam a criptografia fornecida pelo IPSec para proteger sequências de dados durante a sua transferência entre as extremidades do túnel.

Gestão de chaves

O MPPE (*Microsoft Point-to-Point Encryption*), protocolo de nível 2, utiliza uma chave gerada durante a autenticação do utilizador, actualizando-a periodicamente. O IPSec negocia uma chave comum através do ISAKMP e também faz a sua actualização periodicamente.

Suporte a múltiplos protocolos

O tunelamento na camada de ligação de dados suporta múltiplos protocolos, o que facilita o tunelamento de clientes para acesso a redes corporativas utilizando IP, IPX, NetBEUI e outros. Em contraste, os protocolos de tunelamento da camada de rede, tais como o IPSec, suportam apenas redes destino que utilizam o protocolo IP.

2.9.9. Tipos de Túneis

Os túneis podem ser criados de 2 formas diferentes: voluntária e compulsiva:

- Túnel Voluntário - um cliente emite uma solicitação VPN para configurar e criar um túnel voluntário. Neste caso, o computador do utilizador funciona como uma das extremidades do túnel e também como cliente do túnel.
- Túnel Compulsivo - um servidor de acesso VPN *dial-up* configura e cria um túnel compulsivo. Neste caso, o computador do cliente não funciona como extremidade do túnel. Outro dispositivo, o servidor de acesso remoto, localizado entre o computador do utilizador e o servidor do túnel, funciona como uma das extremidades e actua como cliente do túnel.

Tunelamento voluntário

Ocorre quando uma estação ou servidor de encaminhamento utiliza um software de tunelamento cliente para criar uma ligação virtual para o servidor do túnel desejado. O tunelamento voluntário pode requerer ligações IP através de uma LAN ou acesso do tipo *dial-up*.

No caso de acesso do tipo *dial-up*, o mais comum é o cliente estabelecer a ligação antes da criação do túnel. Nas LANs, o cliente já se encontra ligado à rede que pode fornecer o encaminhamento dos dados encapsulados para o servidor de túnel seleccionado. É o caso de clientes que se situem numa LAN corporativa que inicializa túneis para alcançar uma subrede privada na mesma rede.

Tunelamento compulsivo

O computador ou dispositivo de rede que fornece o túnel para o computador cliente é conhecido por diversas formas: FEP (*Front End Processor*) no PPTP, LAC (*L2TP Access Concentrator*) no L2TP ou *IP Security Gateway* no caso do IPSec. Doravante, adoptaremos o termo FEP para denominar esta funcionalidade - ser capaz de estabelecer o túnel quando o cliente remoto se liga [**5**].

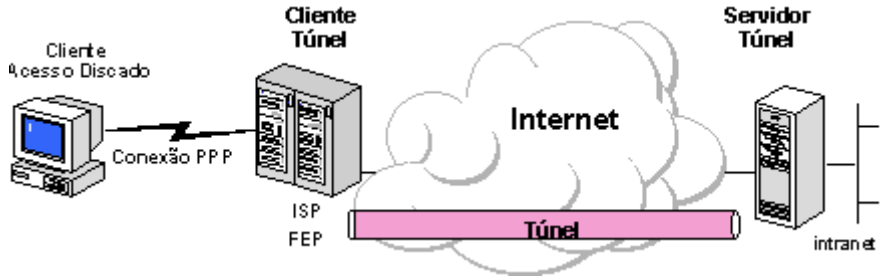


Figura 11 – Tunelamento [5]

No caso da Internet, o cliente faz uma ligação do tipo *dial-up* para um túnel habilitado pelo servidor de acesso no ISP. Por exemplo, uma companhia pode ter um contrato com um ou mais ISPs para disponibilizar um conjunto de FEPs de âmbito nacional. Estas FEPs podem estabelecer túneis sobre a Internet para um servidor de túnel ligado à rede corporativa privada, possibilitando a utilizadores remotos o acesso à rede corporativa através de uma simples ligação local. Esta configuração é conhecida como tunelamento compulsivo, porque o cliente é compelido a usar um túnel criado pelo FEP. Uma vez estabelecida a ligação, todo o tráfego de e para o cliente é automaticamente enviado através do túnel. No tunelamento compulsivo o cliente faz uma ligação PPP. Um FEP pode ser configurado para direcionar todas as ligações *dial-up* para um mesmo servidor de túnel ou, alternativamente, fazer o tunelamento individual baseado na identificação do utilizador ou no destino da ligação.

De forma diferente dos túneis individualizados criados no tunelamento voluntário, um túnel entre o FEP e o servidor de túnel pode ser compartilhado por múltiplos clientes do tipo *dial-up*. Quando um cliente se liga ao servidor de acesso (FEP) e já existe um túnel para o destino desejado, não é necessária a criação de um novo túnel redundante. O próprio túnel existente pode transportar, também, os dados deste novo cliente. No tunelamento compulsivo com múltiplos clientes o túnel só é finalizado no momento em que o último utilizador do túnel se desliga.

2.9.10. IPSEC (Internet Protocol Security) [6]

O IPSec é um protocolo padrão da camada 3, projectado pelo IETF, que oferece transferência segura de informações extremo-a-extremo através de rede IP pública ou privada. Essencialmente, os pacotes IP privados são sujeitos a operações de segurança,

como criptografia, autenticação e integridade, e encapsulados noutros pacotes IP para serem transmitidos. As funções de gestão de chaves também fazem parte das funções do IPSec.

Tal como os protocolos de nível 2, o IPSec trabalha como uma solução para interligação de redes e ligações via linha *dial-up*. Foi projectado para suportar múltiplos protocolos de criptografia, possibilitando que cada utilizador escolha o nível de segurança desejado. Os requisitos de segurança podem ser divididos em 2 grupos, independentes entre si, podendo ser utilizados de forma conjunta ou separada, de acordo com a necessidade de cada utilizador: (i) Autenticação e Integridade; (ii) Confidencialidade. Para implementar estas características, o IPSec é composto por 3 mecanismos adicionais: (i) AH - *Autentication Header*; (ii) ESP - *Encapsulation Security Payload*; (iii) ISAKMP - *Internet Security Association and Key Management Protocol*.

Negociação do nível de segurança

O ISAKMP combina conceitos de autenticação, gestão de chaves e outros requisitos de segurança necessários às transacções e comunicações governamentais, comerciais e privadas na Internet. Com o ISAKMP, as duas máquinas negociam os métodos de autenticação e segurança dos dados, executam a autenticação mútua e geram a chave para criptografar os dados.

Trata-se de um protocolo que rege a troca de chaves criptografadas utilizadas para decifrar os dados. Define procedimentos e formatos de pacotes para estabelecer, negociar, modificar e apagar as SAs (*Security Associations*). As SAs contêm todas as informações necessárias para a execução de serviços variados de segurança na rede, tais como serviços da camada IP (autenticação de cabeçalho e encapsulamento), serviços das camadas de transporte e aplicação ou auto-proteção durante a negociação do tráfego. Também define pacotes para geração de chaves e autenticação de dados. Esses formatos fornecem consistência na transferência de chaves e autenticação de dados de uma forma independente da técnica usada na geração da chave, do algoritmo de criptografia e do mecanismo de autenticação.

O ISAKMP pretende dar suporte para protocolos de segurança em todas as camadas da pilha protocolar. Com a centralização da gestão dos SAs, o ISAKMP minimiza as redundâncias funcionais dentro de cada protocolo de segurança e também pode reduzir o

tempo gasto durante as ligações através da negociação da pilha completa de serviços de uma só vez.

Autenticação e integridade

A autenticação garante que os dados recebidos correspondem àqueles que foram originalmente enviados, assim como garante a identidade do emissor. Integridade significa que os dados transmitidos chegam ao seu destino íntegros, eliminando a possibilidade de terem sido modificados no caminho sem que isso pudesse ser detectado.

O AH é um mecanismo que fornece integridade e autenticação dos datagramas IP. A segurança é garantida através da inclusão de informação para autenticação no pacote, a qual é obtida através de um algoritmo aplicado sobre o conteúdo dos campos do datagrama IP, excluindo-se aqueles que sofrem mudanças durante o transporte. Estes campos abrangem, não só o cabeçalho IP, como todos os outros cabeçalhos e dados do utilizador. No IPv6, o campo *hop-count* e o *time-to-live* (TTL) do IPv4 não são utilizados, pois são modificados ao longo da transferência.

Para alguns utilizadores o uso da autenticação pode ser suficiente, não sendo necessária a confidencialidade.

No IPV6, o AH normalmente é posicionado após os cabeçalhos de fragmentação e *End-to-End* e antes do ESP e dos cabeçalhos da camada de transporte (TCP ou UDP).

Confidencialidade

Propriedade da comunicação que permite que apenas utilizadores autorizados entendam o conteúdo transportado. Desta forma, os utilizadores não autorizados, mesmo tendo capturado o pacote, não poderão ter acesso às informações nele contidas. O mecanismo mais usado para proporcionar esta propriedade é a criptografia.

O serviço que garante a confidencialidade no IPSec é o ESP - *Encapsulating Security Payload*. O ESP também providencia a autenticação da origem dos dados, integridade da ligação e serviço *anti-reply*. A confidencialidade é independente dos demais serviços e pode ser implementada de 2 modos: transporte e túnel. No primeiro modo, o pacote da camada de transporte é encapsulado dentro do ESP e, no túnel, o datagrama IP é encapsulado inteiro dentro do cabeçalho do ESP.

Em suma, as VPNs podem constituir uma alternativa segura para a transmissão de dados através de redes públicas ou privadas, uma vez que já oferecem recursos de autenticação e criptografia com níveis variados de segurança, possibilitando eliminar os *links* dedicados de longa distância e alto custo. Em aplicações onde o tempo de transmissão é crítico, o uso de VPNs através de redes externas ainda deve ser analisado com muito cuidado, pois podem ocorrer problemas de desempenho e atrasos na transmissão sobre os quais a organização não tem nenhum tipo de gestão ou controlo, comprometendo a qualidade desejada nos serviços corporativos.

A decisão de implementar ou não redes privadas virtuais requer uma análise criteriosa dos requisitos, principalmente aqueles relacionados com a segurança, custos, qualidade de serviço e facilidade de uso, que variam de acordo com o negócio de cada organização.

2.10. MPLS/VPN

O modelo de *overlay* VPN, usado frequentemente na rede dos fornecedores de serviço, dita que o planeamento e o estabelecimento de circuitos virtuais através do *backbone* devem ser completamente feitos antes de qualquer tipo de tráfego. No caso de uma rede IP, isto significa que apesar da tecnologia de base não ser orientada à ligação, é requerido um contexto orientado à ligação para fornecer o serviço. Do ponto de vista do fornecedor de serviços, os problemas de escalabilidade do modelo *overlay* VPN são mais sentidos quando é necessário gerir e fornecer um número elevado de circuitos/túneis entre dispositivos de cliente. Do ponto de vista do cliente, o projecto do *Interior Gateway Protocol* é em geral extremamente complexo e de difícil gestão. Por outro lado, o modelo *Peer-to-Peer* VPN sofre da falta de isolamento entre os clientes e da necessidade de coordenação do espaço de endereçamento IP entre eles.

Com a introdução do MPLS, que combina os benefícios do *switching* da camada 2 e do *switching* e *routing* da camada 3, tornou-se possível construir uma tecnologia que combina os benefícios do *overlay* VPN (como segurança e isolamento entre clientes) com os benefícios do encaminhamento simplificado que o *Peer-to-Peer* VPN proporciona.

A nova tecnologia, denominada de MPLS/VPN, resulta num encaminhamento de cliente mais simples e torna possível um número de topologias difíceis de implementar

tanto com o *overlay* como com o *Peer-to-Peer* VPN. O MPLS também acrescenta o benefício da abordagem orientada à ligação ao paradigma do encaminhamento IP através do estabelecimento de caminhos *label-switched* que são criados com base na informação topológica, em vez de fluxo de tráfego.

A arquitectura MPLS/VPN fornece a capacidade de utilizar uma infra-estrutura de rede IP que entrega serviços privados de rede numa infra-estrutura pública. Contudo, os mecanismos para disponibilizar estes serviços são diferentes.

Os serviços VPN são estabelecidos através do uso de *Virtual Routing and Forwarding Instances* (VRFs), onde a informação de routing de cliente de uma VPN específica é introduzida através de mecanismos de importação que utilizam a comunidade estendida do *Route Target* BGP. Esta informação de encaminhamento da VPN é identificada univocamente através do uso de uma distinção de caminho e é distribuída entre os routers fronteira dos fornecedores de serviço, conhecidos como *Provider-Edge Routers* (PE), através do uso de extensões do multi-protocolo BGP.

2.11. Simuladores

Existem diversos simuladores de redes disponíveis no mercado. Como este trabalho assenta na simulação de redes, apresenta-se de seguida uma breve referência comparativa de alguns dos mais conhecidos.

2.11.1. Packet Tracer 5.0 [7]

O Packet Tracer 5.0 oferece uma boa combinação entre um ambiente de simulação realista e uma boa experiência visual, estando disponível sem custo para todos os instrutores da Networking Academy, alunos e ex-alunos.

O Packet Tracer 5.0 corre sobre diversas plataformas, incluindo Windows (Windows XP, Windows 2000, Vista Home Basic e Vista Home Premium) e Linux (Ubuntu e Fedora). Para além disso, suporta uma lista vasta de protocolos que reflecte as tendências actuais das redes, incluindo IPv6, OSPF multi-área, distribuição de rotas, RSTP, SSH e distribuição de chaves em múltiplas camadas.

Este é um software de ensino e aprendizagem. O processo de utilização resume-se basicamente à escrita de instruções, construção de uma rede apropriada ao estudo, especificação de características e simulação final com capacidade de alteração *a posteriori* de todos parâmetros da simulação. O simulador apresenta as *log files* dos dispositivos, bem como os fluxos de pacotes e os seus conteúdos.

O Packet Tracer apresenta essencialmente as seguintes características:

- Ambientes de trabalho lógicos e físicos
- Modos de simulação em tempo real
- Interface do utilizador de fácil utilização
- Lista de eventos global (ferramenta de captura de pacotes)
- Protocolos de LAN, switching, TCP/IP, routing e WAN
- Wizard de actividades e configurador de características laboratoriais
- Suporte de multi-plataformas
- Suporte multi-língua
- Ajuda e tutoriais integrados

Os protocolos suportados pelo Packet Traced 5.0, são:

- HTTP, TFTP, Telnet, SSH, DNS, DHCP;
- TCP and UDP;
- IPv4, ICMP, ARP, IPv6, ICMPv6;
- RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching
- Ethernet (802.3), HDLC, Frame Relay, PPP
- STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP
- 802.11

Este simulador foi desenvolvido com o intuito de ensino académico e é maioritariamente utilizado no complemento do plano de estudos CCNA.

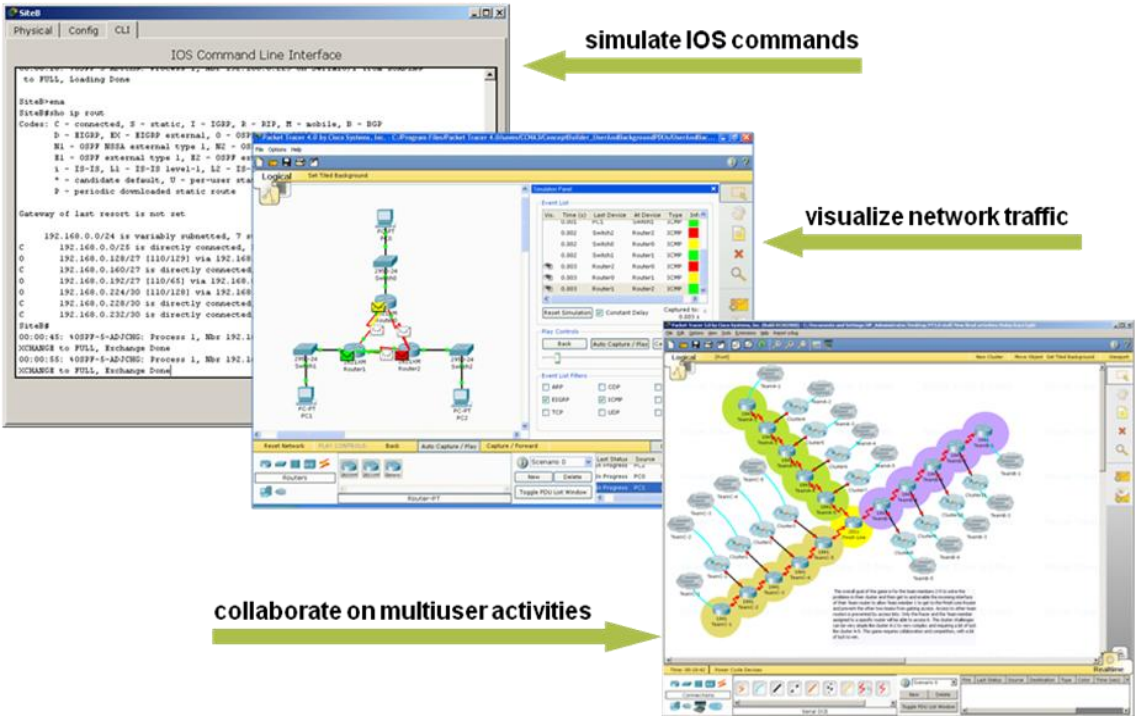


Figura 12 – Simulação, visualização e colaboração no Packet Trace 5.0

2.11.2. NS-2 [8]

NS-2 é um simulador que incorpora diversos protocolos de rede, incluído redes terrestres, wireless e por satélite. Este simulador é bastante popular em meios científicos.

Com este simulador é possível criar diferentes tipologias de rede que incluam diversos algoritmos de encaminhamento, tais como DV, LS, PIM-DM, PIM-SM, AODV e DSR. Permite também simular fontes de tráfego de diversos tipos, tais como WEB, FTP, TELNET, CBR e estocásticas. Possibilita ainda encontrar falhas, incluindo falhas determinísticas, perdas probabilísticas, falhas de ligação, etc. Várias disciplinas de filas de espera (*drop-tail*, RED, FQ, SFQ, DRR, etc.) e de QoS (e.g., *IntServ and Diffserv*) estão também disponíveis. Em termos de visualização é possível seguir o fluxo de pacotes, preenchimento das filas, perda de pacotes, comportamento dos protocolos (TCP *slow start*, *self-clocking*, controlo de congestionamento, retransmissões rápidas e recuperação). Em redes wireless, é possível configurar e visualizar o movimento dos nós. Este simulador tem ainda suporte para MPLS nas suas últimas versões.

2.11.3. OPNET [9]

O OPNET é um simulador que permite o estudo de todas as fases de uma rede: projecto, simulação, captura de dados e análise dos resultados. Disponibiliza um interface gráfico de edição para construir modelos para várias entidades de rede, desde o modelador da camada física a processos de aplicação. Todos são modelados por orientação a objectos, o que confere um mapeamento fácil e intuitivo para sistemas reais. Oferece uma plataforma flexível para testar novas ideias e soluções a baixo custo.

O OPNET é um simulador construído sobre um sistema de eventos discreto, simula o comportamento do sistema modelando cada acontecimento do sistema e faz a computação por processos definidos pelo utilizador. Utiliza uma hierarquia estratégica para organizar todos os modelos para a construção de uma rede inteira. A hierarquia modela entidades, desde transmissores da camada física, antenas, CPUs a correr processos de manuseamento e gestão de filas de espera e protocolos, dispositivos modelados por nós, com modelos de processamento, e transmissores e modelos de redes que interligam todo o tipo de nós. Também fornece ferramentas de programação para definir qualquer tipo de formato de pacote para utilizar nos protocolos criados pelo utilizador. Esta programação inclui a definição do formato de pacote, define o estado da máquina transitória para os processos a correr nos protocolos, define módulos de processamento e de transmissão necessários em cada dispositivo e finalmente define o modelo da rede através da ligação entre dispositivos, utilizando ligações standarizadas ou definidas pelo utilizador.

2.11.4. GNS3 [10]

O GNS3 é um simulador de redes grátis que permite criar topologias de rede, emulando para esse efeito o hardware que se escolhe. Este programa é, no fundo, a junção gráfica de outros projectos:

- Dynamips, um emulador de IOS que permite correr imagens binárias de sistemas Cisco;
- Dynagen, um *front-end* para o Dynamips;
- Pemu, um emulador Pix.

Com o GNS3 pode-se fazer praticamente tudo o que é possível fazer com routers, switch e Pix Cisco reais. Este programa é um emulador e não um simulador, porque o GNS3 utiliza mesmo as imagens binárias dos equipamentos Cisco. Estas imagens (IOS) são o sistema operativo dos equipamentos Cisco.

Assim, se trabalharmos num local que já possua estes equipamentos, podemos testar uma nova versão do IOS antes mesmo de o colocarmos nos equipamentos reais. Se estivermos a estudar, podemos fazê-lo com as últimas versões dos IOS.

O trabalho desta dissertação é realizado neste simulador, principalmente por ser grátis e permitir utilizar os mesmos IOS dos routers, criando um ambiente de simulação que se aproxima bastante do ambiente real.

3. Configuração Básica de MPLS

Neste capítulo e nos próximos serão apresentadas as experiências que foram realizadas, recorrendo ao simulador GNS3, com o objectivo de estudar algumas das características fundamentais do MPLS.

O primeiro cenário proposto é muito simples, sendo constituído por três routers, tal como se apresenta na figura seguinte. Este será o cenário base das diferentes experiências que serão realizadas nas próximas secções.

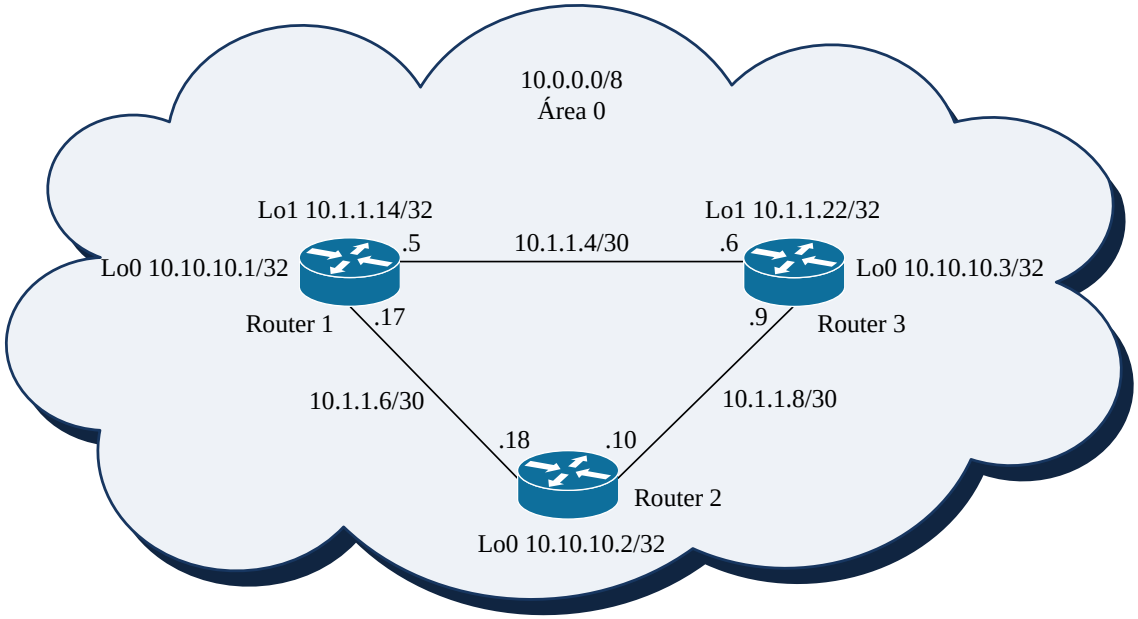


Figura 13 – Rede MPLS base

Para a realização destas experiências, tendo como base o cenário da Figura 13, utilizou-se o simulador GNS3 e consideraram-se routers CISCO 7200 a correr o IOS "c7200-adventerprisek9-mz.124-4.T1". A captura de pacotes foi efectuada recorrendo ao WireShark. Em todas as experiências descritas nos próximos sub-capítulos foram utilizados os mesmos endereços IP e os mesmos endereços de *loopback*.

3.1. Configuração Básica de MPLS utilizando o protocolo OSPF

Nesta sub-secção apenas se programaram nos routers os protocolos base que servem de apoio ao MPLS, para além naturalmente dos endereços dos interfaces reais e de *loopback*. Seguidamente, activou-se o protocolo OSPF e o Cisco Expedited Forwarding

(CEF) e por fim o MPLS com LDP. Nesta experiência pretendem-se estudar apenas os pacotes que são trocados no processo de configuração da rede durante a activação do MPLS. Os comandos utilizados na programação dos routers são os seguintes (por exemplo, para o Router 1):

```
>enable
>configure terminal
>interface f0/1
>ip address 10.1.1.17 255.255.255.252
>no shutdown
>interface f0/0
>ip address 10.1.1.5 255.255.255.252
>no shutdown
>interface loopback 0
>ip address 10.10.10.1 255.255.255.255
>no shutdown
>interface loopback 1
>ip address 10.1.1.14 255.255.255.255
>no shutdown
>end
>write
>conf t
>ip cef
>do show running-config interface f0/0 | include cef no ip route-cache cef
>interface f0/0
>ip route-cache cef
>router ospf 1
>network 10.1.1.4 0.0.0.3 area 0
>network 10.1.1.16 0.0.0.3 area 0
>end
>conf t
>mpls ldp router-id loopback 0
>interface f0/0
>mpls ip
>interface f0/1
>mpls ip
>end
>write
```

Para os restantes routers as configurações são as mesmas, diferindo apenas nos endereços IP. Durante e após a configuração dos routers foram capturados os pacotes em algumas das interfaces dos routers, recorrendo ao analisador de protocolos WireShark. Os pacotes que apresentamos na Figura 14 foram capturados na interface f0/0 do Router 1.

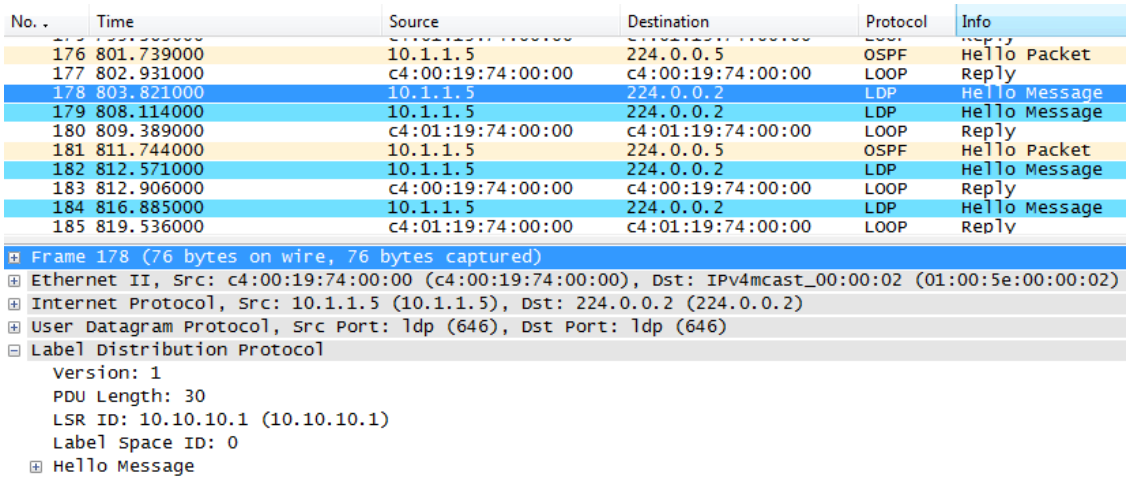


Figura 14 – Captura de pacotes na interface f0/0 do Router1 no início da activação do MPLS.

Nesta captura podemos observar, por exemplo, que o pacote nº178 é um pacote LDP correspondente a uma mensagem *Hello*. É neste tipo de pacote que se constata a utilização de etiquetas MPLS. Neste caso a informação contida na etiqueta será apenas a informação contida nas mensagens *Hello*, isto porque como ainda estamos no início da activação do protocolo MPLS o pacote 178 é enviado em multicast para toda a rede. Ou seja, constatamos que existem etiquetas a circular na rede mas, como ainda estamos no início do processo de convergência, inda não se observa uma utilização expressa das etiquetas.

3.2. Configuração Básica MPLS utilizando OSPF e MPLS-TE

Nesta experiência alteraram-se as configurações dos routers e, em vez de se utilizar MPLS com LDP, utilizou-se o MPLS-TE, uma vez que este permitirá posteriormente a criação de túneis, o assunto que será tratado na secção seguinte. Esta experiência serve portanto de base para a experiência que será proposta na próxima secção. Assim, nesta secção descrevem-se os comandos necessários para a activação do MPLS-TE, passando pela programação dos endereços IP, de endereços loopback, e pela activação do protocolo RSVP.

Tomando como exemplo as configurações efectuadas no router 3, foram então inseridos os seguintes comandos para configurar os endereços IP e activar o protocolo OSPF:

```
>enable
>configure terminal
>interface f0/1
```

```
>ip address 10.1.1.9 255.255.255.252
>no shutdown
>interface f0/0
>ip address 10.1.1.6 255.255.255.252
>no shutdown
>interface loopback 0
>ip address 10.10.10.3 255.255.255.255
>no shutdown
>interface loopback 1
>ip address 10.1.1.22 255.255.255.255
>no shutdown
>end
>write
>conf t
>ip cef
>do show running-config interface f0/0 | include cef no ip route-cache cef
>interface f0/0
>ip route-cache cef
>router ospf 1
>network 10.1.1.4 0.0.0.3 area 0
>network 10.1.1.8 0.0.0.3 area 0
>end
>conf t
>router ospf 1
>mpls traffic-eng area 0
>mpls traffic-eng router-id Lo0
>end
>write
```

Seguidamente, activou-se o protocolo RSPV:

```
>conf t
>mpls traffic-eng tunnels
>interface f0/0
>mpls traffric-eng tunnels
>interface f0/1
>mpls traffic-eng tunnels
>end
>conf t
>interface f0/0
>ip rsvp bandwidth 512 512
>interface f0/1
>ip rsvp bandwidth 512 512
>end
>write
```

Após a programação e activação de todos os protocolos, realizou-se exactamente a mesma programação nos restantes routers, tendo apenas em conta os respectivos endereços IP.

3.3. Criação de Túneis

Após a programação dos endereços e activação dos protocolos, passa-se agora à criação de túneis. Existem dois tipos de túneis possíveis de serem criados: estáticos e dinâmicos. Cada túnel pode ainda conter uma série de propriedades que podem ser programadas, tais como, a prioridade, o *path option*, a largura de banda e o *load-share*, por exemplo.

Nesta secção pretende-se mostrar como se devem programar os routers de modo a criar túneis MPLS. Assim sendo, vão ser criados dois túneis com rotas estáticas, diferindo apenas na prioridade, e dois túneis com rotas dinâmicas, que irão diferir nas prioridades e nas larguras de banda disponíveis. Nas secções seguintes realizar-se-á uma análise mais cuidadosa e pormenorizada de cada uma das propriedades que é possível configurar nos túneis MPLS.

Os túneis estáticos iniciam-se no router 3 e têm como destino o router 1, passando um deles pelo router 2 enquanto que o outro liga directamente o router 3 ao router 1. Os túneis dinâmicos iniciam-se no router 1 e têm como destino o router 3.

No Router 3 utilizam-se os seguintes comandos:

```
>conf t
>interface tunnel 1
> ip unnumbered loopback 0
>tunnel destination 10.10.10.1
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel mpls traffic-eng priority 2 2
>tunnel mpls traffic-eng bandwidth 150
>tunnel mpls traffic-eng path-option 1 explicit name low
>end
>conf t
>interface tunnel 2
>ip unnumbered loopback 0
>tunnel destination 10.10.10.1
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel traffic-eng priority 4 4
>tunnel mpls traffic-eng bandwidth 200
>tunnel mpls traffic-eng path-option 1 explicit name straight
>end
>conf t
>ip explicit-path name low enable
>next-address 10.1.1.10
>next-address 10.1.1.17
>ip explicit-path name straight enable
>next-address 10.1.1.5
>end
>write
```

No Router 1 são introduzidos os seguintes comandos:

```
>conf t
>interface tunnel 3
>ip unnumbered loopback 0
>no ip directed-broadcast
>tunnel destination 10.10.10.3
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel mpls traffic-eng priority 5 5
>tunnel mpls traffic-eng bandwidth 100
>tunnel mpls traffic-eng path-option 2 dynamic
>end
>conf t
>interface tunnel 4
>ip unnumbered loopback 0
>no ip directed-broadcast
>tunnel destination 10.10.10.3
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel mpls traffic-eng priority 6 6
>tunnel mpls traffic-eng bandwidth 60
>tunnel mpls traffic-eng path-option 1 dynamic
>end
>write
```

Durante a programação acima referida, capturaram-se os seguintes pacotes na interface f0/0 do router 3:

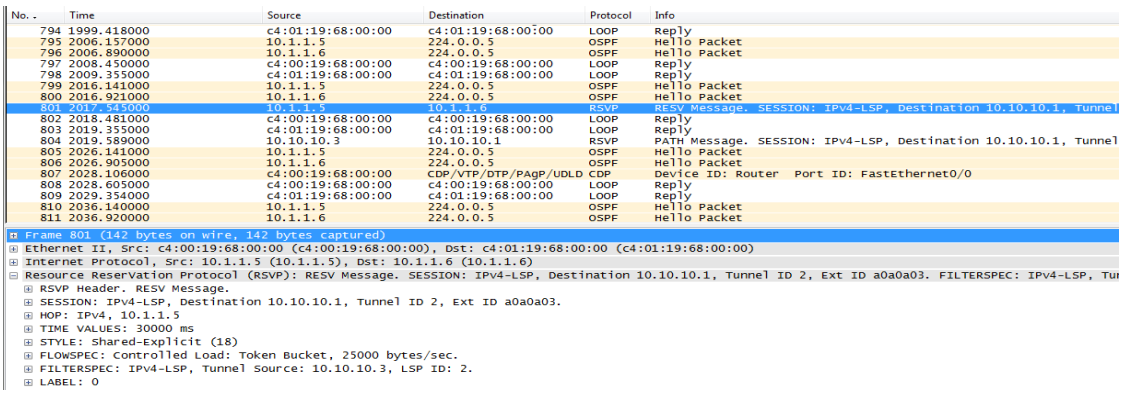


Figura 15 – Captura na interface f0/0 do Router3, efectuada durante o estabelecimento dos túneis.

A partir da observação da Figura 15, pode-se afirmar que as etiquetas MPLS são encapsuladas nos pacotes RSVP (ex. pacote 801). Da análise do pacote 801, verifica-se que dentro do campo RSVP e no objecto SESSION é referido o protocolo IPv4-LSP. Como o protocolo de transporte deixou de ser o LDP, as etiquetas são agora encapsuladas no protocolo RSVP, já que é este o protocolo que foi configurado para atender às necessidades do MPLS-TE.

3.4. Comparação dos diferentes parâmetros dos túneis

De modo a comparar o efeito dos diferentes parâmetros dos túneis, começou-se por criar dois túneis que se iniciam no Router 3 e terminam no Router 1, ligando ambos os routers de forma directa. Numa tentativa de estudar o efeito do parâmetro prioridade, as únicas diferenças entre os túneis residem nas suas prioridades. Assim, o túnel 1 apresenta uma prioridade de 2 e o túnel 2 uma prioridade de 4. Isto significa que à partida e, de acordo com o que é sabido, quanto menor for o valor colocado no campo da prioridade maior será a prioridade do túnel. Deste modo, programaram-se os routers de acordo com os comandos descritos no quadro seguinte.

Router 3:

```
>conf t
>interface tunnel 1
> ip unnumbered loopback 0
>tunnel destination 10.10.10.1
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel traffic-eng priority 2 2
>tunnel mpls traffic-eng bandwidth 150
>tunnel mpls traffic-eng path-option 1 explicit name low
>end
>conf t
>interface tunnel 2
>ip unnumbered loopback 0
>tunnel destination 10.10.10.1
>tunnel mode mpls traffic-eng
>tunnel mpls traffic-eng autoroute announce
>tunnel traffic-eng priority 4 4
>tunnel mpls traffic-eng bandwidth 150
>tunnel mpls traffic-eng path-option 1 explicit name fast
>end
>conf t
>ip explicit-path name low enable
>next-address 10.1.1.5
>ip explicit-path name fast enable
>next-address 10.1.1.5
>end
>write
```

Após os routers estarem devidamente programados e os túneis criados efectuaram-se *pings* para os endereços 10.1.1.5, 10.1.1.17, 10.1.1.18 e 10.1.1.10 com o objectivo de, com o auxílio do comando *show interface tunnel 1 accounting*, visualizar o número de pacotes que passam por cada um dos túneis, verificando dessa forma por que túneis passam os

pacotes correspondentes aos *pings* realizados. Assim, na Tabela 3 apresentam-se os resultados obtidos.

Ping	Túnel
10.1.1.5	x
10.1.1.17	2
10.1.1.18	1
10.1.1.10	x

Tabela 3 – Resultados correspondentes aos pings realizados para o estudo da Prioridade (1)

Antes de se retirar qualquer conclusão, optou-se por inverter agora as prioridades dos túneis de modo a verificar-se se o funcionamento se altera de algum modo. Desta forma o procedimento executado foi exactamente o mesmo procedimento anterior apenas com a diferença de que agora a prioridade do túnel 1 foi alterada para 7 e a do túnel 2 se manteve em 4. Na Tabela 4 apresentam-se os resultados obtidos:

Ping	Túnel
10.1.1.5	x
10.1.1.17	2
10.1.1.18	1
10.1.1.10	x

Tabela 4 – Resultados correspondentes aos pings realizados para o estudo da Prioridade (2)

Mediante a observação da Tabela 3 e da Tabela 4 a única afirmação que pode ser realizada de imediato é de que os resultados obtidos até agora, no que diz respeito à influência da prioridade, são absolutamente inconclusivos. Mais a frente retomar-se-á o estudo desta característica de uma forma pormenorizada. Optou-se por protelar esta análise uma vez que neste momento ainda não conseguimos perceber qual dos parâmetros Prioridade ou *Path-Option* tem uma influência maior na escolha dos túneis por parte dos routers MPLS.

Uma vez que existe uma grande variedade de parâmetros que podem ser estudados relativamente aos túneis MPLS, optou-se por estudar-se agora o parâmetro *Path Option*. Para este estudo, mantiveram-se exactamente os mesmos túneis que foram criados para o estudo da prioridade, alterando-se agora as Prioridades de ambos os túneis para 4 de modo

a que esse parâmetro não interfira nos resultados obtidos e alterou-se no túnel 1 o *Path Option* para 2, mantendo-se o *Path Option* do túnel 2 em 1. O objectivo será que o caminho escolhido seja agora o do túnel 2, uma vez que este apresenta o *Path Option* mais baixo, ou seja, aquele que deve ser atendido prioritariamente. O *Path Option* não é mais do que um nível de opção por um determinado caminho, já que por vezes podem existir vários caminhos alternativos que conduzam ao mesmo destino: o *Path Option* permite que esses caminhos sejam utilizados de uma forma ordenada, se não existir nenhum impedimento, permitindo assim um maior controlo do fluxo de informação na rede.

Router 3:

```
>conf t
>interface tunnel 1
>tunnel traffic-eng priority 4 4
>tunnel mpls traffic-eng path-option 2 explicit name low
>no tunnel mpls traffic-eng path-option 1 explicit name low
>end
>write
```

Após as alterações efectuadas nos routers e depois de se terem novamente efectuado os *pings* para os endereços 10.1.1.5, 10.1.1.17, 10.1.1.18 e 10.1.1.10, obtiveram-se os seguintes resultados (Tabela 5).

Ping	Túnel
10.1.1.5	X
10.1.1.17	2
10.1.1.18	1
10.1.1.10	x

Tabela 5 – Resultados referentes aos pings efectuados para o estudo do parâmetro *Path Option*

Da observação da Tabela 5 continua a não ser possível tirar qualquer tipo de conclusão imediata, uma vez que os resultados obtidos não estão muito de acordo com o que era esperado.

Mesmo com resultados inconclusivos, existem duas situações que devem ser testadas antes de se partir para uma análise mais pormenorizada: devem ser testadas as situações que temos diferentes prioridades e diferentes *Path Option* e ainda a situação em

que os *Path Option* apresentam caminhos distintos. Para a primeira situação, programaram-se os routers de modo a que (i) o túnel 1 apresente uma prioridade igual a 2 e um *Path Option* igual a 1 e o túnel 2 apresente uma prioridade igual a 4 e um *Path Option* de 2; (ii) foi também testada a situação inversa em que o túnel 1 apresenta uma prioridade de 2 e um *Path Option* de 2 e o túnel 2 uma prioridade de 4 e um *Path Option* de 1.

Assim, no Router 3 foi feita seguinte configuração:

```
>conf t
>interface tunnel 1
>tunnel mpls traffic-eng priority 2 2
>end
>write
```

Ping	Túnel
10.1.1.5	X
10.1.1.17	2
10.1.1.18	1
10.1.1.10	X

Tabela 6 – Pings efectuados no estudo de diferentes *Path Option* e diferentes prioridades (situação i)

Ping	Túnel
10.1.1.5	X
10.1.1.17	2
10.1.1.18	1
10.1.1.10	X

Tabela 7 – Pings efectuados no estudo de diferentes *Path Option* e diferentes prioridades (situação ii)

Para analisar a última situação foi necessário alterar algumas características: criou-se um caminho que liga o Router 3 ao Router 1 mas passando agora pelo Router 2. Este caminho foi atribuído ao túnel 2, que manteve uma prioridade de 4, enquanto que o túnel 1 foi mantido com o caminho que tem sido utilizado até agora em todos os exercícios e que liga directamente o Router 3 ao Router 1, mantendo também a prioridade igual a 2.

No router Router 3 foram então efectuadas as seguintes configurações:

```
>conf t
>interface tunnel 1
 >tunnel traffic-eng priority 2 2
>no tunnel mpls traffic-eng path-option 2 explicit name low
>tunnel mpls traffic-eng path-option 1 explicit name low
>interface tunnel 2
```

Os resultados obtidos depois de mais uma vez se efectuarem os pings para os endereços 10.1.1.5, 10.1.1.17, 10.1.1.18 e 10.1.1.10, estão representados na tabela seguinte:

Ping	Túnel
10.1.1.5	X
10.1.1.17	2
10.1.1.18	1
10.1.1.10	X

Tabela 8 – Resultados referentes aos pings realizados para o estudo de diferentes caminhos

Analisando os resultados apresentados nas tabelas 6, 7 e 8, podemos mais uma vez afirmar que estes resultados não são muito coerentes e inclusivamente não permitem tirar nenhum conclusão. De tal forma, que agora resta apenas e agora tentar perceber de uma forma mais pormenorizada o que correu mal ao longo deste exercício. Assim sendo, nas próximas secções os parâmetros Prioridade e *Path Option* vão ser estudados de uma forma mais pormenorizada, deixando assim quaisquer conclusões sobre o seu funcionamento para mais tarde.

3.5. Estudo do parâmetro Prioridade

Para estudarmos o Priority de uma forma simples e verificarmos o seu correcto funcionamento, montamos o cenário da Figura 13, colocando ligações série entre os routers de modo a limitar o tráfego e conseguir verificar de forma inequívoca a maneira como os túneis são escolhidos. Assim sendo, no início do estudo vão existir apenas 3 túneis (Figura 16):

Túnel 1- directo entre o router 3 e o router 1

Túnel 2- directo entre o router 3 e o router 1

Túnel 3- entre o router 3 e o router 1 com passagem pelo router 2.

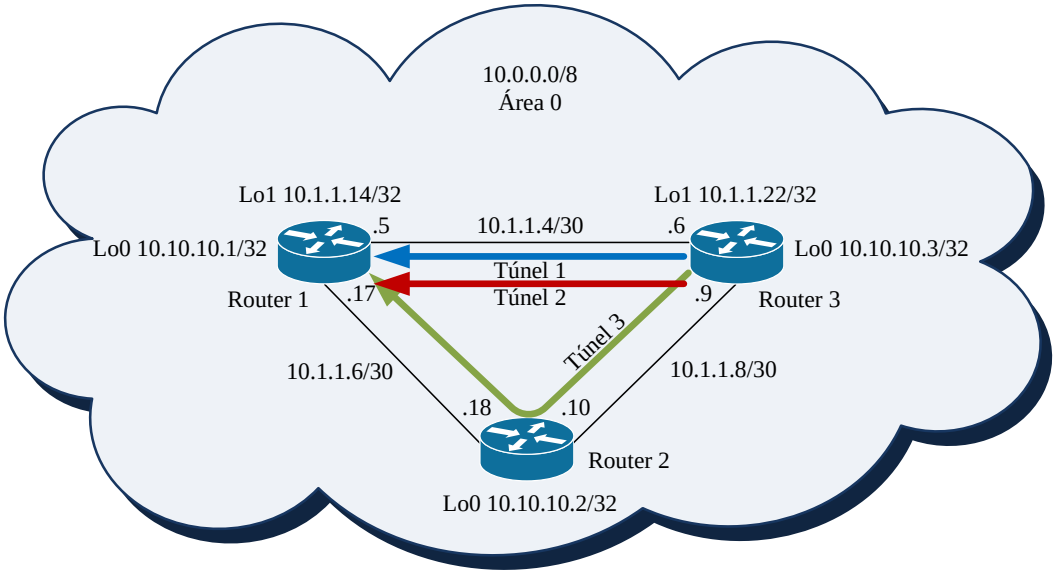


Figura 16 – Cenário base para o estudo da Prioridade: Representação dos Túneis utilizados

3.5.1. Cenário 1

No primeiro cenário utilizam-se apenas os túneis 1 e 2, com as seguintes características:

Túnel 1: Prioridade - 1 1; Path Option - 1

Túnel 2: Prioridade - 2 2; Path Option – 1

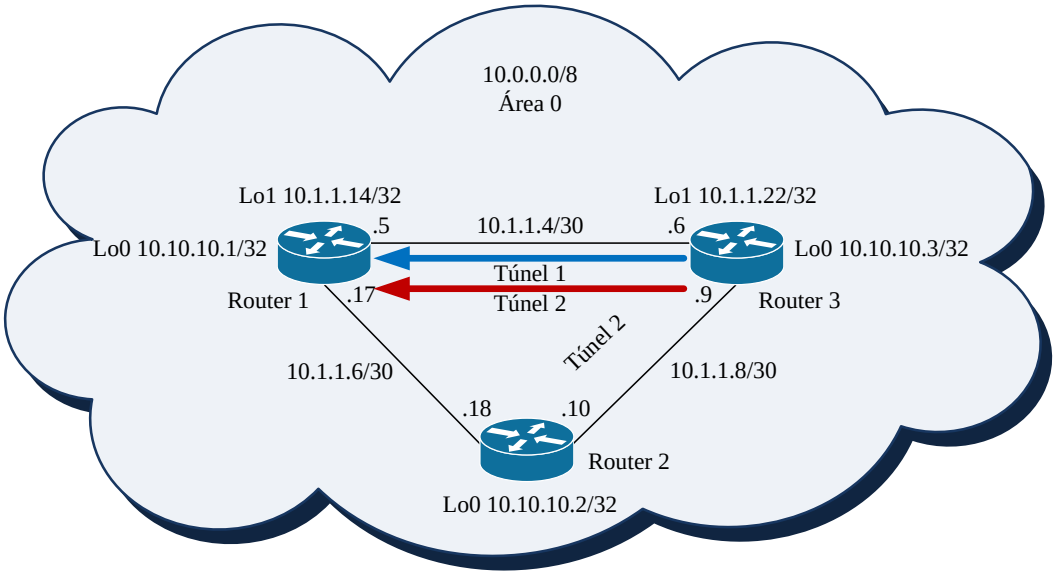


Figura 17 – Estudo da Prioridade: Utilização de apenas dos Túneis 1 e 2

Ou seja, o túnel 1 tem Prioridade 1 e o túnel 2 tem Prioridade 2, sendo que ambos possuem o mesmo *Path-Option*. Recorrendo novamente ao comando *sh interface tunnel X accounting*, onde X corresponde ao número do túnel, e ao comando *ping*, verificou-se por que túneis fluíam os pacotes resultantes. No quadro apresentam-se os resultados obtidos e toda a informação disponibilizada pelos routers

```
Router#sh mpls traffic-eng tunnel

Name: Router_t1                    (Tunnel1) Destination: 10.10.10.1
  Status:
    Admin: up        Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit tun1 (Basis for Setup, path weight 64)

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial1/0, implicit-null
  RSVP Signalling Info:
       Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 1, Tun_Instance 124
    RSVP Path Info:
      My Address: 10.10.10.3
      Explicit Route: 10.1.1.5 10.10.10.1
      Record Route:  NONE
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  History:
    Tunnel:
      Time since created: 57 minutes, 50 seconds
      Time since path change: 1 minutes, 19 seconds
    Current LSP:
      Uptime: 1 minutes, 20 seconds
    Prior LSP:
      ID: path option 1 [50]
      Removal Trigger: path error

Name: Router_t2                    (Tunnel2) Destination: 10.10.10.1
  Status:
    Admin: up        Oper: down    Path: not valid   Signalling: Down
    path option 1, type explicit tun2

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 2  2   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
auto-bw: disabled
```

Identificação doRouter e a Rota do Túnel definida neste.

```
  History:
    Tunnel:
      Time since created: 53 minutes, 12 seconds
      Time since path change: 1 minutes, 37 seconds
    Prior LSP:
      ID: path option 1 [321]
      Removal Trigger: path error
      Last Error: PCALC:: Can't use link 0.0.0.0 on node 10.10.10.3

Name: Router_t3                         (Tunnel3) Destination: 10.10.10.1
  Status:
    Admin: admin-down Oper: down   Path: not valid   Signalling: Down
    path option 1, type explicit tun3

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
    auto-bw: disabled

  History:
    Tunnel:
      Time since created: 47 minutes, 37 seconds
```

```
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         30       3000
Router#sh interface tunnel 2 accounting
Tunnel2
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         47       4088
```

Contador do número de pacotes que passam no Túnel 1

Contador do número de pacotes que passam no Túnel 2.

```
Router#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/168/252 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         35       3500
Router#sh interface tunnel 2 accounting
Tunnel2
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         47       4088
Router#ping 10.1.1.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/103/168 ms
```

```
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        40       4000
Router#sh interface tunnel 2 accounting
Tunnel2
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        47       4088
Router#ping 10.1.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/215/484 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        45       4500
Router#sh interface tunnel 2 accounting
Tunnel2
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        47       4088
```

Como podemos observar pelos resultados acima apresentados, a informação é sempre enviada pelo túnel 1, já que este túnel tem o menor valor de *Priority* (quanto menor foi o *Priority* maior será a prioridade do túnel). Apenas é estabelecido o túnel 1.

3.5.2. Cenário 2

Nesta experiência trocaram-se as prioridades dos túneis, isto é, o túnel 1 passou a ter prioridade igual a 2 e o túnel 2 prioridade igual a 1. Deste modo, este exercício vai permitir verificar se as prioridades dos túneis estão realmente a funcionar, já que é esperado que os pacotes gerados pelo comando *ping* circulem agora pelo túnel 2, uma vez que este apresenta a o valor mais baixo de prioridade (ou seja, é mais prioritário). Assim, no quadro seguinte são apresentados todos os resultados obtidos:

Túnel 1: Prioridade - 2 2; *Path Option* - 1

Túnel 2: Prioridade - 1 1; *Path Option* - 1

```
Router#sh mpls traffic-eng tunnel

Name: Router_t1                           (Tunnel1) Destination: 10.10.10.1
  Status:
    Admin: up         Oper: down   Path: not valid   Signalling: Down
    path option 1, type explicit tun1
```

```
  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 2  2   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
    auto-bw: disabled

  History:
    Tunnel:
      Time since created: 49 minutes, 18 seconds
      Time since path change: 26 minutes, 56 seconds
    Prior LSP:
      ID: path option 1 [50]
      Removal Trigger: path error
      Last Error: PCALC:: Can't use link 0.0.0.0 on node 10.10.10.3

Name: Router_t2                   (Tunnel2) Destination: 10.10.10.1
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit tun2 (Basis for Setup, path weight 64)
```

Informação sobre o Túnel 1.Neste caso interessa observar a prioridade definida, que é 2.

```
  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial1/0, implicit-null
  RSVP Signalling Info:
       Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 2, Tun_Instance 235
    RSVP Path Info:
      My Address: 10.10.10.3
      Explicit Route: 10.1.1.5 10.10.10.1
      Record Route:  NONE
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
Record Route:  NONE
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  History:
    Tunnel:
      Time since created: 45 minutes, 1 seconds
      Time since path change: 27 minutes, 41 seconds
    Current LSP:
      Uptime: 27 minutes, 42 seconds
    Prior LSP:
      ID: path option 1 [234]
      Removal Trigger: configuration changed

Name: Router_t3                   (Tunnel3) Destination: 10.10.10.1
  Status:
    Admin: admin-down Oper: down    Path: not valid   Signalling: Down
    path option 1, type explicit tun3

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
```

Informação sobre o Túnel 2. Neste caso interessa observar a prioridade definida, que é 1.

AutoRoute: enabled LockDown: disabled Loadshare: 200 bw-based
auto-bw: disabled

History:
Tunnel:
Time since created: 39 minutes, 33 seconds

Router#sh int tunnel 1 accounting
Tunnel1

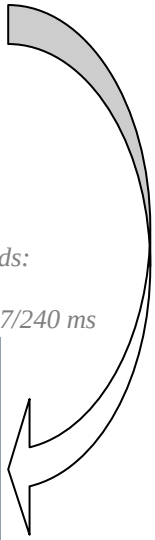
	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	30	3000

Router#sh int tunnel 2 accounting
Tunnel2

	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	32	2588

Router#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/117/240 ms



Aumento do contador do número de pacotes que passam no Túnel 2

Router#sh int tunnel 1 accounting
Tunnel1

	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	30	3000

Router#sh int tunnel 2 accounting
Tunnel2

	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	37	3088

Router#ping 10.1.1.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/115/184 ms

Router#sh int tunnel 1 accounting
Tunnel1

	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	30	3000

Router#sh int tunnel 2 accounting
Tunnel2

	Protocol	Pkts In	Chars In	Pkts Out	Chars Out
	IP	0	0	42	3588

Router#ping 10.1.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/149/260 ms

```
Router#sh int tunnel 1 accounting
Tunnel1
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        30       3000
Router#sh int tunnel 2 accounting
Tunnel2
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        47       4088
```

Como se pode observar dos resultados acima apresentados, o tráfego gerado é totalmente encaminhado pelo túnel 2, já que este tem o menor valor de prioridade (é mais prioritário).

3.5.3. Cenário 3

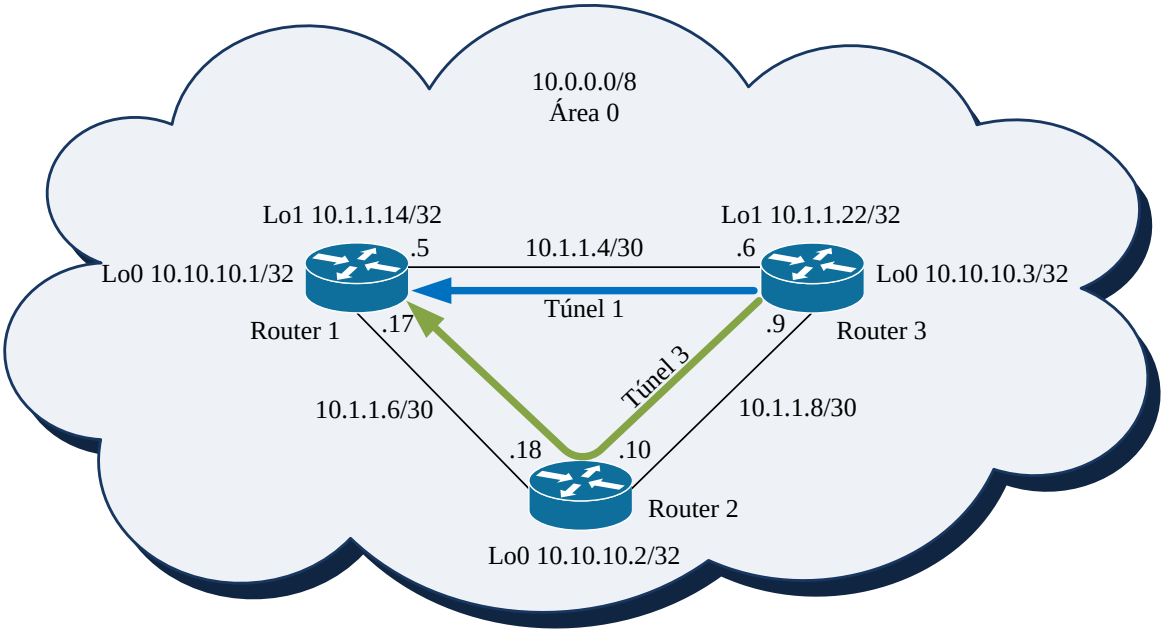


Figura 18 – Estudo da Prioridade: Utilização de apenas dos Túneis 1 e 3

Neste cenário vamos utilizar dois túneis que apresentam caminhos diferentes: o túnel 1 tem um valor de Priority igual a 1 e o túnel 3 tem um valor de Priority igual a 2. Mais uma vez é esperado que o tráfego gerado pelo comando *ping* circule pelo túnel com menor valor de Priority. No quadro seguinte apresentam-se os resultados desta experiência.

Túnel 1: Prioridade - 1 1; *Path Option* - 1

Túnel 3: Prioridade - 2 2; *Path Option* – 1

Router#sh mpls traffic-eng tunnel

Name: Router_t1 (Tunnel1) Destination: 10.10.10.1
Status:
Admin: up Oper: up Path: valid Signalling: connected

path option 1, type explicit tun1 (Basis for Setup, path weight 64)

Config Parameters:
Bandwidth: 200 kbps (Global) Priority: 1 1 Affinity: 0x0/0xFFFF
Metric Type: TE (default)
AutoRoute: enabled LockDown: disabled Loadshare: 200 bw-based
auto-bw: disabled

InLabel : -
OutLabel : Serial1/0, implicit-null
RSVP Signalling Info:
Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 1, Tun_Instance 124
RSVP Path Info:
My Address: 10.10.10.3
Explicit Route: 10.1.1.5 10.10.10.1
Record Route: NONE
Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
History:
Tunnel:
Time since created: 1 hours, 6 minutes
Time since path change: 10 minutes
Current LSP:
Uptime: 10 minutes, 1 seconds
Prior LSP:
ID: path option 1 [50]
Removal Trigger: path error

Name: Router_t3 (Tunnel3) Destination: 10.10.10.1
Status:
Admin: up Oper: down Path: not valid Signalling: Down

path option 1, type explicit tun3

Config Parameters:
Bandwidth: 200 kbps (Global) Priority: 2 2 Affinity: 0x0/0xFFFF
Metric Type: TE (default)
AutoRoute: enabled LockDown: disabled Loadshare: 200 bw-based
auto-bw: disabled

History:
Tunnel:
Time since created: 56 minutes, 36 seconds
Path Option 1:

```
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         45       4500
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
Router#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/120/316 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         50       5000
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
Router#ping 10.1.1.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/180/300 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         55       5500
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
Router#ping 10.1.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/164/316 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   IP         0          0         60       6000
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
```

Como se observa pelos resultados acima apresentados, o tráfego é sempre enviado em todos os casos pelo túnel com o menor valor de prioridade.

3.5.4. Cenário 4

Nesta experiência, e continuando a lógica que foi utilizada nos cenários anteriores, inverteram-se mais uma vez as prioridades dos túneis, isto é, o túnel 1 agora apresenta um valor de Priority de 2 e o túnel 3 apresenta um valor de Priority igual a 1. Espera-se então que o tráfego flua pelo o túnel de menor valor de Priority (mais prioritário). Seguem no quadro os resultados obtidos.

Túnel 1: Prioridade - 2 2; *Path Option* - 1

Túnel 3: Prioridade - 1 1; *Path Option* – 1

```
Router#sh mpls traffic-eng tunnel

Name: Router_t1                    (Tunnel1) Destination: 10.10.10.1
  Status:
    Admin: up        Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit tun1 (Basis for Setup, path weight 64)

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 2  2   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial1/0, implicit-null
  RSVP Signalling Info:
       Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 1, Tun_Instance 127
    RSVP Path Info:
      My Address: 10.10.10.3
      Explicit Route: 10.1.1.5 10.10.10.1
      Record Route:  NONE
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  History:
    Tunnel:
      Time since created: 1 hours, 12 minutes
      Time since path change: 1 minutes, 8 seconds
    Current LSP:
      Uptime: 1 minutes, 8 seconds
    Prior LSP:
      ID: path option 1 [124]
      Removal Trigger: configuration changed

Name: Router_t3                    (Tunnel3) Destination: 10.10.10.1
  Status:
    Admin: up        Oper: up     Path: valid       Signalling: connected
```

```
  path option 1, type explicit tun3 (Basis for Setup, path weight 128)

 Config Parameters:
  Bandwidth: 200      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
  auto-bw: disabled

 InLabel  :  -
 OutLabel : Serial1/1, 16
 RSVP Signalling Info:
     Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 3, Tun_Instance 34
  RSVP Path Info:
    My Address: 10.10.10.3
    Explicit Route: 10.1.1.10 10.1.1.17 10.10.10.1
    Record Route:  NONE
    Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  RSVP Resv Info:
    Record Route:  NONE
    Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
 History:
  Tunnel:
    Time since created: 1 hours, 5 minutes
    Time since path change: 44 seconds
  Current LSP:
    Uptime: 44 seconds
Router#sh interface tunnel 1 accounting
Tunnel1
            Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                 IP         0         0         60       6000
Router#sh interface tunnel 3 accounting
Tunnel3
            Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
Router#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/142/256 ms
Router#sh interface tunnel 1 accounting
Tunnel1
            Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                 IP         0         0         65       6500
Router#sh interface tunnel 3 accounting
Tunnel3
            Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
Router#ping 10.1.1.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/120/200 ms
```

```
Router#sh interface tunnel 1 accounting
Tunnel1
          Protocol  Pkts In  Chars In  Pkts Out  Chars Out
               IP        0         0        65       6500
Router#sh interface tunnel 3 accounting
Tunnel3
          Protocol  Pkts In  Chars In  Pkts Out  Chars Out
               IP        0         0         5        500
Router#ping 10.1.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/225/352 ms
Router#sh interface tunnel 1 accounting
Tunnel1
          Protocol  Pkts In  Chars In  Pkts Out  Chars Out
               IP        0         0        65       6500
Router#sh interface tunnel 3 accounting
Tunnel3
          Protocol  Pkts In  Chars In  Pkts Out  Chars Out
               IP        0         0        10       1000
```

Como podemos observar, o tráfego é sempre enviado pelo túnel com menor valor de prioridade.

Em suma, para que o campo Priority tenha um correcto funcionamento temos de ter em conta o tipos de ligações físicas existentes entre os routers, pois como se pôde observar na secção 3.4 o parâmetro Prioridade não apresentava um funcionamento dentro do previsto. Tal facto devia-se à utilização de ligações físicas Ethernet, uma vez que estas apresentam uma largura de banda bastante elevada. Assim, as imposições de limitação de tráfego necessárias acabavam por não ser respeitadas. Nesta subsecção, como se utilizaram ligações série, o funcionamento do parâmetro Prioridade foi o esperado. Assim, sempre que se pretenda que um túnel seja utilizado preferencialmente deve-se colocar um valor de Priority menor do que é definido para os outros túneis, garantindo assim a utilização preferencial deste túnel por parte dos routers. Contudo, como já tinha sido referido, existem outras características que influenciam a escolha de um túnel para o envio de informação. Nas próximas secções essas características serão estudadas em detalhe.

3.6. Estudo do parâmetro *Path Option*

Como o próprio nome indica, o parâmetro *Path-Option* estabelece caminhos opcionais para o encaminhamento de tráfego numa rede MPLS em que tenham sido

definidos túneis. A hierarquia de prioridades na utilização de cada um dos caminhos definidos é estabelecida pelo gestor da rede aquando do estabelecimento desses caminhos. Assim, a probabilidade de utilização de um dos caminhos varia em função dos requisitos que foram tidos em conta pelo gestor. O facto desta definição ser estática faz com que o encaminhamento possa não levar em conta a proximidade física ou outra medida de custo, como por exemplo o número de ligações que o tráfego percorre. A lista de caminhos opcionais não implica que todo o tráfego tome a opção mais prioritária, ou seja:

1. Na situação em que o volume de dados excede as capacidades definidas no túnel será também utilizado o *Path-Option* que precede o *Path-Option* actual e assim consecutivamente, até ser possível processar todo o fluxo de dados.
2. É senso comum que deve existir redundância numa rede, pelo que o gestor de rede deve estabelecer diferentes túneis por forma a implementar diversas opções de caminho entre dois pontos na rede. Assim, na eminência de uma quebra numa ligação física que faça parte da definição de um túnel, o túnel tornar-se-á inutilizável e o *Path-Option* ficará inválido.

Sem a definição do *Path-Option* a qualidade de túneis estáticos não é estabelecida, pelo que a sua definição é obrigatória para pelo menos um *Path-Option*.

Para o estudo então do *Path-Option* vamos utilizar o cenário representado na Figura 19.

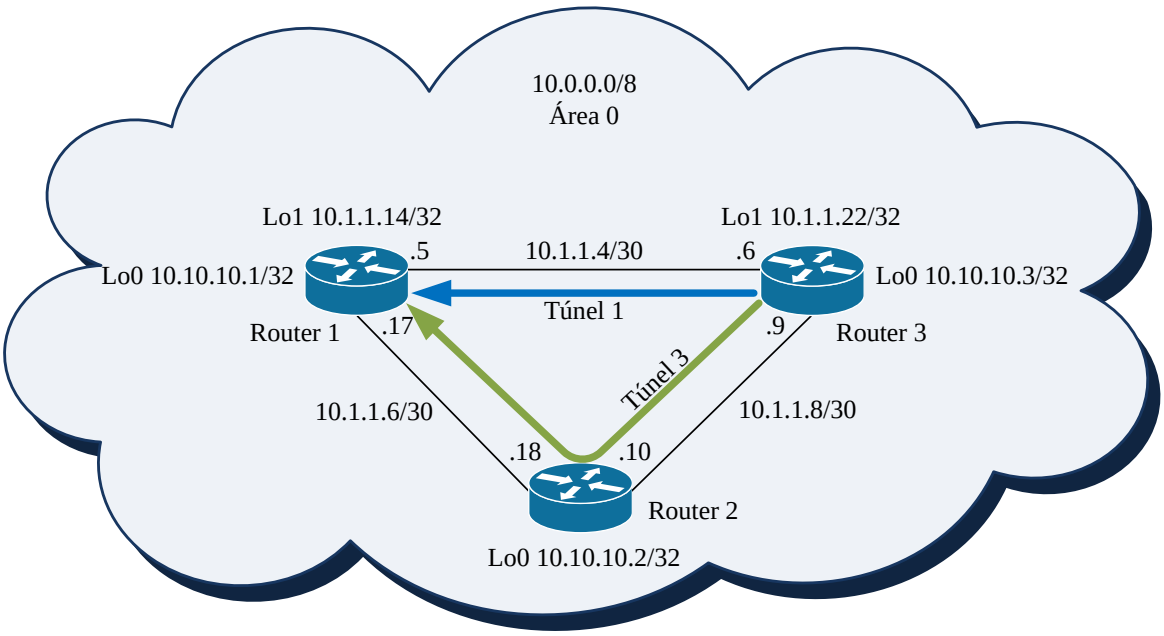


Figura 19 – Cenário para estudo do parâmetro *Path Option*

3.6.1. Cenário 1

No cenário representado na Figura 19 foram utilizados os túneis 1 e 3 (os mesmos túneis que foram utilizados no estudo da Prioridade). O túnel 1 tem uma Prioridade de 1 e um *Path-Option* de 1, sendo o caminho entre o Router 3 e o Router 1 directo; o túnel 3 tem uma Prioridade de 1 e um *Path-Option* de 2, sendo o caminho definido aquele que liga o Router 3 ao Router 1 passando pelo Router 2. Através da utilização do comando *ping*, realizado para diferentes endereços, verifica-se que túnel é utilizado no envio dos pacotes, utilizando o comando *sh interface tunnel X accounting*. Desta forma, apresentam-se no quadro seguinte todos os comandos utilizados e as respostas dadas aos routers.

```
  path option 1, type explicit tun1 (Basis for Setup, path weight 64)

 Config Parameters:
  Bandwidth: 200     kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
  auto-bw: disabled

 InLabel  :  -
 OutLabel : Serial1/0, implicit-null
 RSVP Signalling Info:
     Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 1, Tun_Instance 128
  RSVP Path Info:
    My Address: 10.10.10.3
    Explicit Route: 10.1.1.5 10.10.10.1
    Record Route:  NONE
    Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  RSVP Resv Info:
    Record Route:  NONE
    Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
 History:
  Tunnel:
    Time since created: 1 hours, 56 minutes
    Time since path change: 37 minutes, 46 seconds
  Current LSP:
    Uptime: 37 minutes, 46 seconds
    Selection: reoptimation
  Prior LSP:
    ID: path option 1 [127]
    Removal Trigger: configuration changed
Name: Router_t3                   (Tunnel3) Destination: 10.10.10.1
 Status:
  Admin: up        Oper: up     Path: valid       Signalling: connected
  path option 2, type explicit tun3 (Basis for Setup, path weight 128)

 Config Parameters:
  Bandwidth: 200     kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute:  enabled   LockDown: disabled  Loadshare: 200      bw-based
```

Informação sobre o Túnel 1

Informação sobre o Túnel 1. Rota explicita definida.

Informação sobre o Túnel 3

auto-bw: disabled

InLabel : -
OutLabel : Serial1/1, 16
RSVP Signalling Info:
Src 10.10.10.3, Dst 10.10.10.1, Tun_Id 3, Tun_Instance 40
RSVP Path Info:
My Address: 10.10.10.3
Explicit Route: 10.1.1.10 10.1.1.17 10.10.10.1
Record Route: NONE

Informação sobre o Túnel 3. Rota explícita definida.

Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
History:
Tunnel:
Time since created: 1 hours, 46 minutes
Time since path change: 1 minutes, 6 seconds
Current LSP:
Uptime: 1 minutes, 6 seconds
Prior LSP:
ID: path option 1 [39]
Removal Trigger: path option removed

Router#sh interface tunnel 1 accounting
Tunnel1

Protocol	Pkts In	Chars In	Pkts Out	Chars Out
IP	0	0	92	8576

Router#sh interface tunnel 3 accounting
Tunnel3

Protocol	Pkts In	Chars In	Pkts Out	Chars Out
IP	0	0	32	2576

Router#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/115/188 ms

Router#sh interface tunnel 1 accounting
Tunnel1

Protocol	Pkts In	Chars In	Pkts Out	Chars Out
IP	0	0	97	9076

Router#sh interface tunnel 3 accounting
Tunnel3

Protocol	Pkts In	Chars In	Pkts Out	Chars Out
IP	0	0	32	2576

Router#ping 10.1.1.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/161/244 ms

```
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        97       9076
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        37       3076
Router#ping 10.1.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/190/324 ms
Router#sh interface tunnel 1 accounting
Tunnel1
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        97       9076
Router#sh interface tunnel 3 accounting
Tunnel3
              Protocol  Pkts In  Chars In  Pkts Out  Chars Out
                    IP        0         0        42       3576
```

Neste quadro verifica-se que ambos os túneis são criados e a informação que por eles passa varia consoante o endereço de destino dos pacotes: quando o ping é realizado para o endereço 10.10.10.1, é utilizado o túnel 1 mas quando o ping é realizado para os endereços 10.1.1.17 ou 10.1.1.18 já é utilizado o túnel 3. Tal facto deve-se não só a existir largura de banda suficiente para a criação de ambos os túneis, como também ao facto do *priority* ser igual em ambos os túneis, porque neste caso os routers não atendem ao *Path-Option* mas sim ao Priority, ou seja, o campo *Path-Option* só é entendido quando este é utilizado dentro do mesmo túnel.

3.6.2. Cenário 2

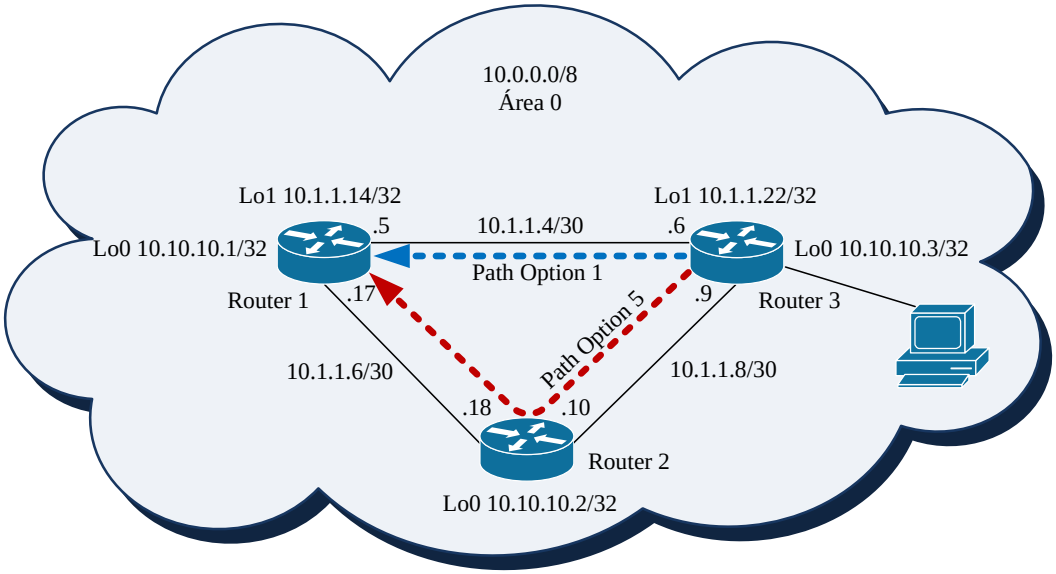


Figura 20 – Estudo do Path Option: Um Túnel dois caminhos

Neste cenário aboliu-se um dos túneis e passou-se a trabalhar apenas com um túnel e dois *Path-Options (*Figura 20*)*. Assim, manteve-se o *Path-Option* 1 e o criou-se um *Path-Option* 5 , em que o caminho definido é o que se inicia no Router 3 e termina no Router 1 passando pelo Router 2. Realizaram-se alguns pings e efectuaram-se capturas com o WireShark de modo a conseguir-se distinguir porque caminho estão a ser enviados os pacotes.

No.	Time	Source	Destination	Protocol	Info
8	10.546000	N/A	N/A	CDP	Device ID: Router
9	10.640000	10.1.1.6	224.0.0.5	OSPF	Hello Packet
10	10.796000	N/A	N/A	SLARP	Line keepalive, outg
11	11.810000	10.1.1.5	224.0.0.5	OSPF	Hello Packet
12	11.950000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
13	12.090000	10.10.10.1	10.10.10.3	ICMP	Echo (ping) reply
14	12.215000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
15	12.402000	10.10.10.1	10.10.10.3	ICMP	Echo (ping) reply
16	12.512000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
17	12.590000	10.10.10.1	10.10.10.3	ICMP	Echo (ping) reply
18	12.605000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
19	12.761000	10.10.10.1	10.10.10.3	ICMP	Echo (ping) reply
20	12.824000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
21	12.839000	10.10.10.1	10.10.10.3	ICMP	Echo (ping) reply
22	19.937000	N/A	N/A	SLARP	Line keepalive, outg
23	20.561000	10.1.1.6	224.0.0.5	OSPF	Hello Packet

```
⊞ Frame 13 (104 bytes on wire, 104 bytes captured)
⊞ Cisco HDLC
⊞ Internet Protocol, Src: 10.10.10.1 (10.10.10.1), Dst: 10.10.10.3 (10.10.10.3)
⊟ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0 ()
    Checksum: 0xd38d [correct]
    Identifier: 0x0021
    Sequence number: 0 (0x0000)
  ⊞ Data (72 bytes)
```

Figura 21 – Captura entre o Router 3 e o Router 1

Conforme se pode observar na Figura 21, os pacotes são enviados pelo *Path-Option* 1, já que foi apenas no link entre o Router 3 e o Router 1 que se capturaram pacotes. Tal facto vai de encontro ao que era esperado, pois o Router, desde que tenha recursos disponíveis, opta sempre pelo menor *Path-Option*.

3.6.3. Cenário 3

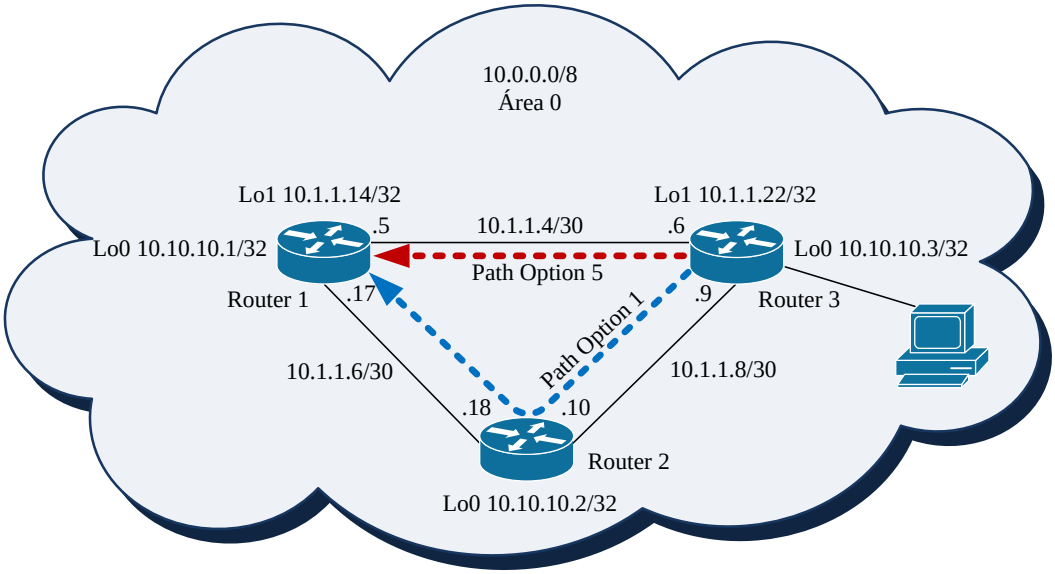


Figura 22 – Estudo do Path Option: Path Options invertidos

Nesta experiência, e tendo como base a experiência anterior, inverteu-se o *Path-Option*, isto é, alterou-se o número do *Path-Option* do caminho que passa pelo Router 2 de 5 para 1 e alterou-se o *Path-Option* do caminho directo entre o Router 3 e o Router 1 de 1 para 5 (Figura 22). Assim sendo, é esperado que o tráfego gerado pelos *pings* vá passar todo pelo caminho que liga o Router 3 ao Router 1 passando pelo Router 2. Efectuaram-se capturas entre os *links* que ligam o Router 3 ao Router 2 e o Router 3 ao Router 1. Na Figura 23 e na Figura 24 apresentam-se as capturas efectuadas.

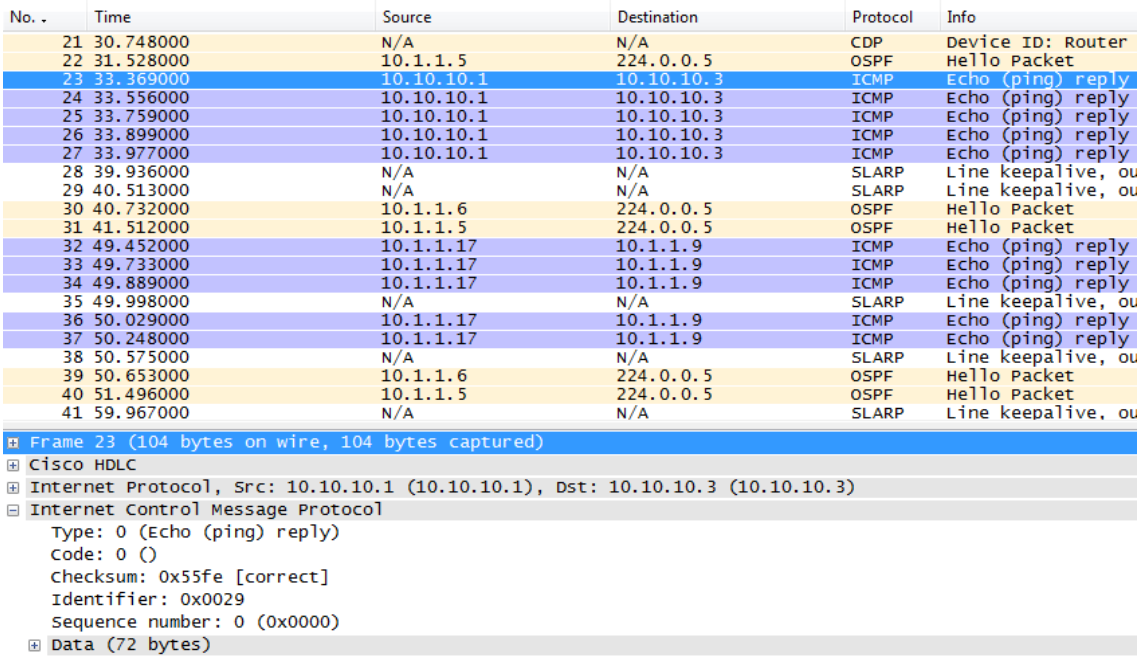


Figura 23 – Captura entre o Router 3 e o Router 1

Mediante a observação de ambas as figuras, pode-se concluir que o tráfego gerado é todo enviado pelo caminho mais longo, uma vez que tal como era esperado o tráfego é todo enviado pelo caminho que tem o menor *Path-Option*. Observando com atenção a Figura 24 verifica-se que os pacotes desta captura são apenas *ICMP Echo Request*, já os pacotes ICMP Echo Reply se encontram todos na captura efectuada na ligação entre o Router 3 e o Router 1 (Figura 23). Isto significa que os pacotes são realmente enviados pelo caminho com menor *Path-Option*, mas as respostas não são reencaminhadas obrigatoriamente pelo mesmo caminho.

No.	Time	Source	Destination	Protocol	Info
6	6.739000	10.1.1.9	224.0.0.5	OSPF	Hello Packet
7	8.814000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
8	9.079000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
9	9.282000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
10	9.438000	10.1.1.10	224.0.0.5	OSPF	Hello Packet
11	9.484000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
12	9.609000	10.10.10.3	10.10.10.1	ICMP	Echo (ping) request
13	15.646000	N/A	N/A	SLARP	Line keepalive, outg
14	16.458000	N/A	N/A	SLARP	Line keepalive, outg
15	16.520000	10.1.1.9	224.0.0.5	OSPF	Hello Packet
16	19.500000	10.1.1.10	224.0.0.5	OSPF	Hello Packet
17	24.913000	10.1.1.9	10.1.1.17	ICMP	Echo (ping) request
18	25.272000	10.1.1.9	10.1.1.17	ICMP	Echo (ping) request
19	25.459000	10.1.1.9	10.1.1.17	ICMP	Echo (ping) request
20	25.677000	10.1.1.9	10.1.1.17	ICMP	Echo (ping) request
21	25.677000	N/A	N/A	SLARP	Line keepalive, outg
22	25.786000	10.1.1.9	10.1.1.17	ICMP	Echo (ping) request
23	26.457000	N/A	N/A	SLARP	Line keepalive, outg
24	26.520000	10.1.1.9	224.0.0.5	OSPF	Hello Packet
25	29.421000	10.1.1.10	224.0.0.5	OSPF	Hello Packet
26	31.761000	10.1.1.10	10.1.1.9	RSVP	RESV Message. SESSIO
27	35.646000	N/A	N/A	SLARP	Line keepalive, outg
28	36.473000	N/A	N/A	SLARP	Line keepalive, outg
29	36.535000	10.1.1.9	224.0.0.5	OSPF	Hello Packet
30	39.452000	10.1.1.10	224.0.0.5	OSPF	Hello Packet
31	40.123000	10.1.1.9	10.1.1.18	ICMP	Echo (ping) request
32	40.295000	10.1.1.18	10.1.1.9	ICMP	Echo (ping) reply
33	40.357000	10.1.1.9	10.1.1.18	ICMP	Echo (ping) request
34	40.466000	10.1.1.18	10.1.1.9	ICMP	Echo (ping) reply
35	40.513000	10.1.1.9	10.1.1.18	ICMP	Echo (ping) request
36	40.685000	10.1.1.18	10.1.1.9	ICMP	Echo (ping) reply
37	40.700000	10.1.1.9	10.1.1.18	ICMP	Echo (ping) request
38	40.841000	10.1.1.18	10.1.1.9	ICMP	Echo (ping) reply
39	40.997000	10.1.1.9	10.1.1.18	ICMP	Echo (ping) request
40	41.137000	10.1.1.18	10.1.1.9	ICMP	Echo (ping) reply

Figura 24 – Captura entre o Router 3 e o Router 2

3.6.4. Cenário 4

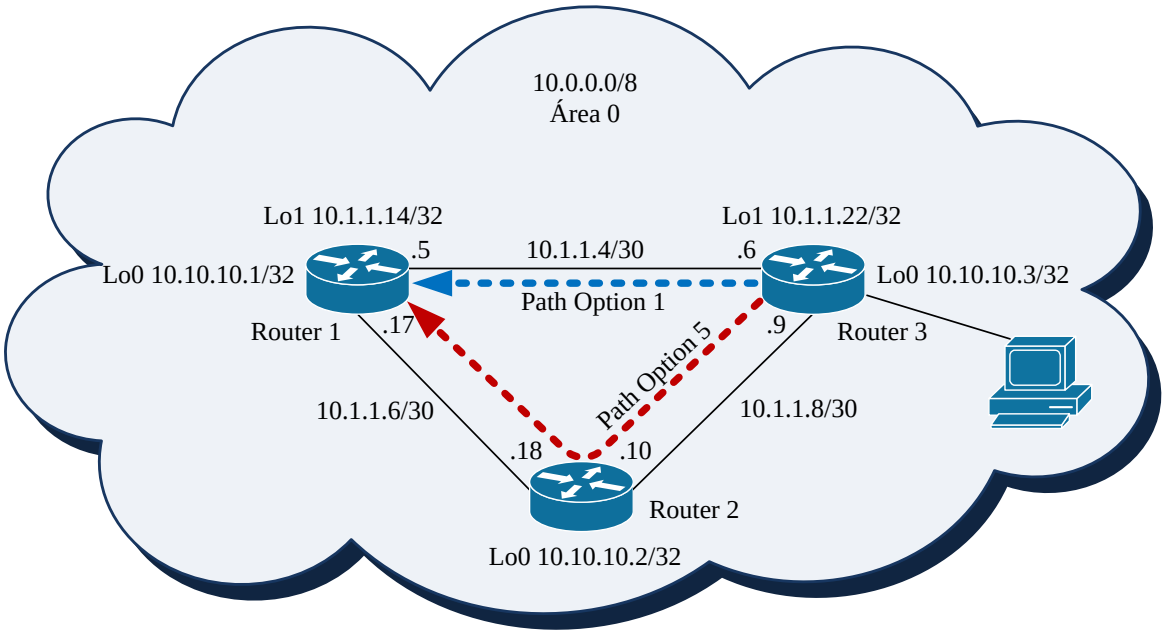


Figura 25 – Estudo do Path Option: Utilização de um gerador de Tráfego

Nesta fase do estudo sobre o *Path Option*, falta agora testar a passagem de forma automática de um caminho para outro, dentro do mesmo túnel, ou seja, quando o caminho

que estiver a ser utilizado estiver sobrecarregado de tráfego, este deverá passar a ser enviado também pela segunda opção de caminho disponível e programada no túnel. Esta passagem deve ocorrer de forma automática. Para se poder testar esta opção, tem que se garantir que as ligações vão ficar sobre sobrecarregadas com tráfego. Como tal, vamos acrescentar à nossa rede um gerador de tráfego. Assim colocou-se um terminal ligado ao Router 3 (com o endereço IP 10.2.2.1) e, mediante a utilização do software de geração de tráfego IPERF, gerou-se o tráfego necessário para que as ligações ficassem absolutamente sobrecarregadas (Figura 25).

Neste cenário temos então um túnel com dois caminhos, um com *Path Option* igual a 1 (caminho "tun3", que liga o router 3 ao router 1 com passagem pelo router 2) e outro com *Path Option* igual a 5 (caminho "tun1", que liga o router 3 ao router 1 de forma directa). No IPERF colocou-se a seguinte linha de código:

iperf –c 10.10.10.1 –u. (Gera tráfego do tipo UDP)

Realizou-se uma captura (recorrendo ao programa WireSharrk) no Router 1 de modo a observar que caminho(s) estava(m) a ser utilizado(s) pelo túnel por forma a fazer a distribuição de carga.

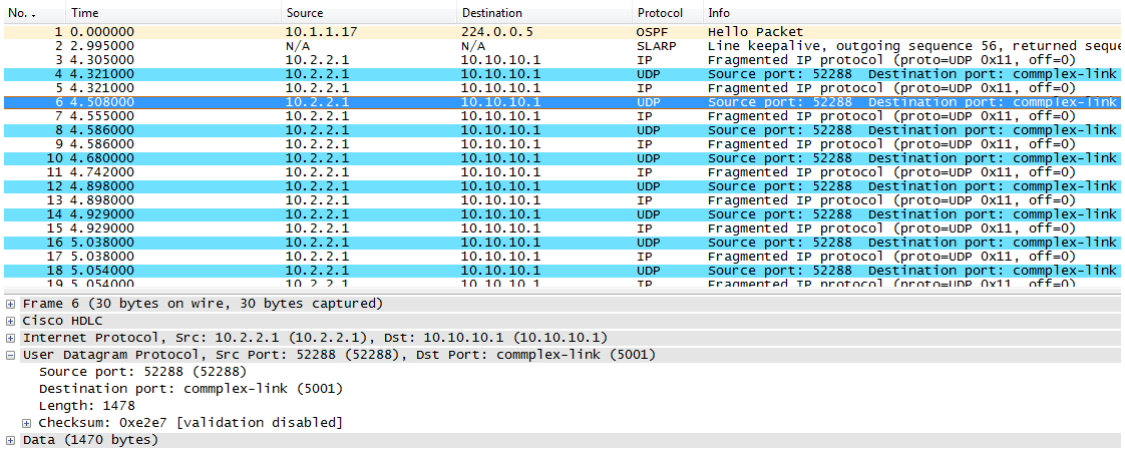


Figura 26 – Captura no Router 1

Através da observação da Figura 26, verifica-se que o porto de envio é sempre o mesmo, o que significa que todo o tráfego gerado foi enviado pelo mesmo caminho. Contudo, uma vez que o tráfego gerado pelo IPERF consistiu apenas num fluxo de pacotes, não foi possível observar a passagem de um caminho para outro de forma automática, como era pretendido.

3.6.5. Cenário 5

Uma vez que o resultado obtido na subsecção anterior não foi exactamente o pretendido, possivelmente devido à maneira como o tráfego foi gerado pelo IPERF, optou-se agora por gerar mais do que um fluxo de tráfego. Assim, neste cenário mantiveram-se os mesmos caminhos no mesmo túnel (caminho tun3 e tun1). No IPERF foram geradas as seguintes linhas de *código*

iperf –c 10.10.10.1 –u –t 120 –i 0.5 (Gera tráfego do tipo UDP, com uma duração de 120 segundos com um intervalo de 0.5 segundos)

iperf –c 10.10.10.1 –u –p 5005 –t 60 –i 1.

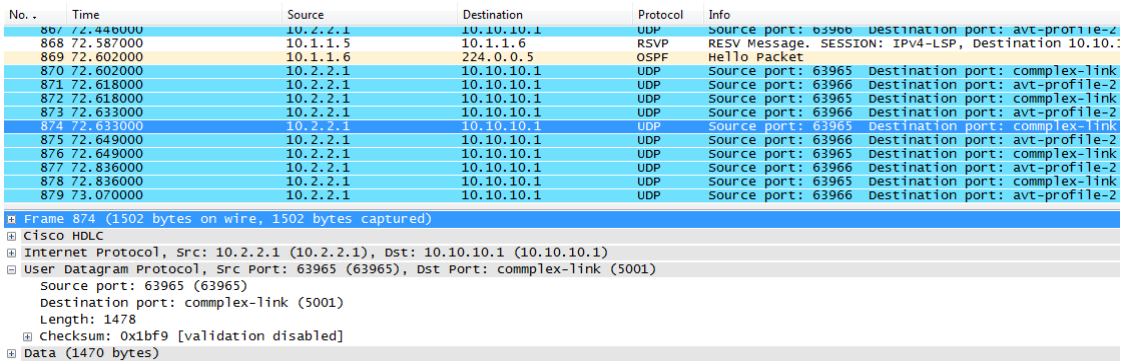


Figura 27 – Captura entre o Router 3 e o Router 1

Através da observação da captura ilustrada na Figura 27, realizada recorrendo mais uma vez ao WireShark, verifica-se que os resultados obtidos não são de todo os esperados. Em primeiro lugar, continua a não haver distribuição de uma parte do tráfego pelo segundo caminho, uma vez que teoricamente o primeiro caminho não suporta todo o tráfego que por ele está a ser enviado. Em segundo lugar, o fluxo de pacotes gerado está a passar por um único caminho, por sinal o caminho com o maior *Path Option*, o que não deveria acontecer. Assim sendo, os resultados obtidos nesta experiência não são de todo os esperados: em primeiro lugar, seria de esperar que o tráfego passasse pelo caminho de menor *Path Option*, e em segundo lugar, que quando o primeiro caminho estivesse sobrelotado se passasse para o segundo caminho disponível no túnel.

3.7. Uma nova abordagem ao parâmetro *Path Option*

De modo a poder estudar-se novamente a passagem entre os diferentes caminhos presentes num mesmo túnel, acrescentou-se ao cenário representado na Figura 19 o router R4, ligando os Routers R1 (10.1.1.41) e R3 (10.1.1.34). O Router R4 possui um interface loopback 0 cujo endereço IP é 10.10.10.4 e os endereços IP das interfaces que ligam aos routers R1 e R3 são 10.1.1.42 e 10.1.1.33, respectivamente.

O objectivo do estudo passa agora por, tendo mais do que um caminho a ligar o Router R3 ao Router R1, se cortarem as ligações durante a transmissão dos pacotes de modo a verificar se os routers optam por outros caminhos que se encontrem disponíveis. Esta escolha seguirá em principio a hierarquia do *Path-Option*, isto é, quanto maior for o valor de *Path-Option* menor será a prioridade do respectivo caminho.

Assim sendo, definiram-se três caminhos distintos dentro do túnel 1:

Caminho 1: Entre o Router 3 e o Router 1, passando pela ligação directa.

Caminho 2: Entre o Router 3 e o Router 1, passando pelo Router 4.

Caminho 3: Entre o Router 3 e o Router 1, passando pelo Router 2.

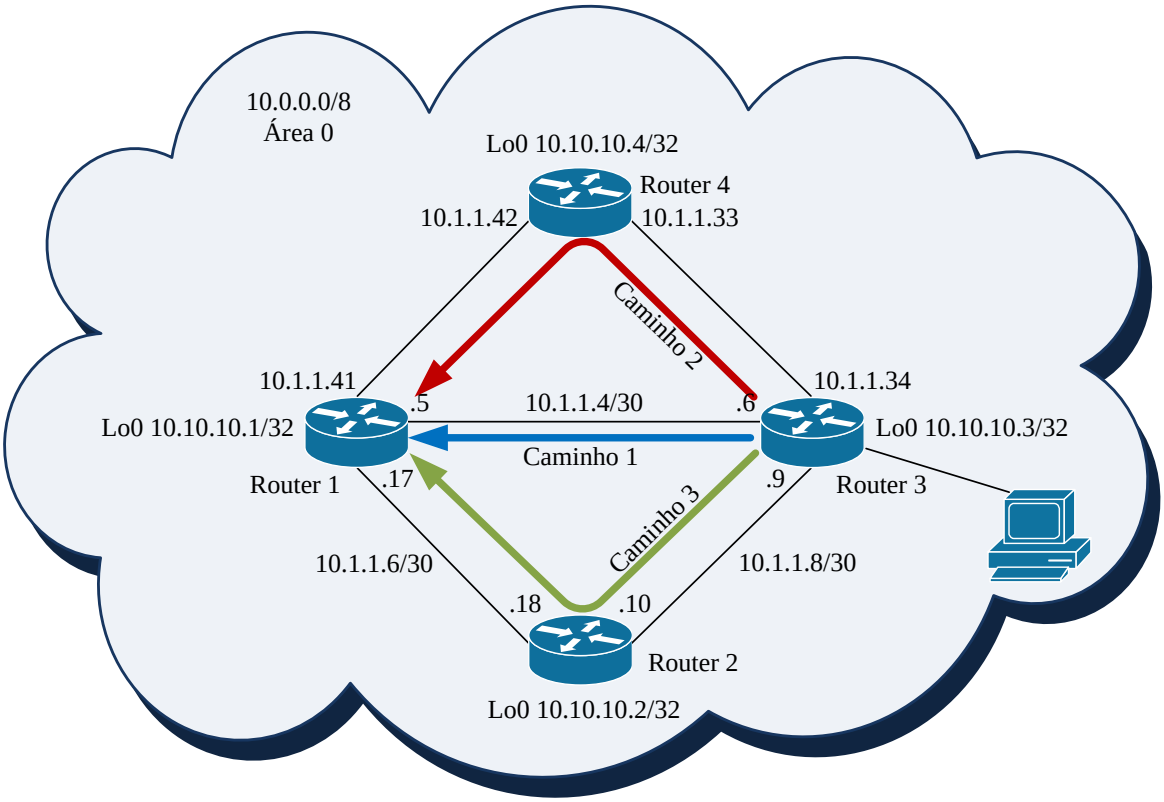


Figura 28 – Cenário 2 para o estudo do *Path Option*

3.7.1. Cenário 1

Programaram-se então no Router 3 os seguintes valores de *Path-Option*:

- Path-Option1: caminho 1
- Path-Option3: caminho 2
- Path-Option5: caminho 3

Criaram-se 4 fluxos de pacotes semelhantes no IPERF, que foram enviados para o endereço 10.10.10.1. A linha de comando introduzida no IPERF foi *iperf –c 10.10.10.1 –u –t 120 –i 1 –b 100000*. Obteve-se a captura de pacotes apresentada na Figura 29:

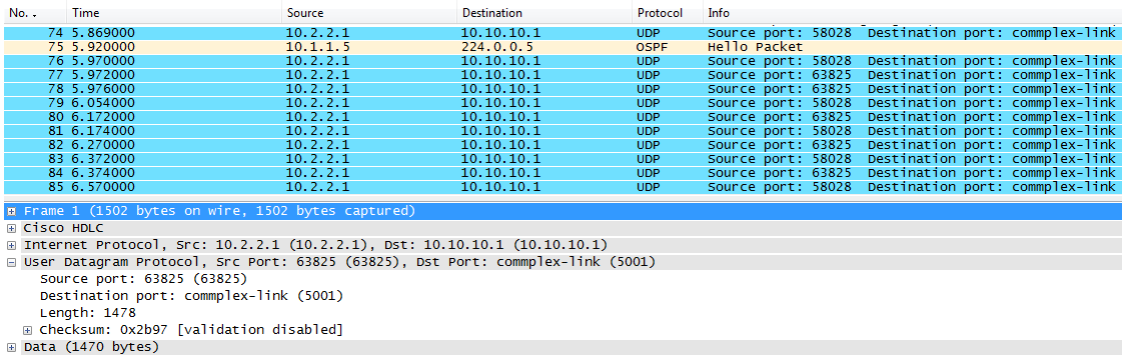


Figura 29 – Captura entre o Router 3 e o Router 1

Observando a Figura 29, verifica-se que todo o tráfego passa pelo caminho 1, tal como era esperado já que este caminho apresentava o menor *Path-Option*, não existindo partilha de tráfego pelos outros caminhos disponíveis nos túneis.

3.7.2. Cenário 2

Neste exercício seguiu-se exactamente a mesma estratégia do exercício anterior, isto é, mantiveram-se os três caminhos com os diferentes *Path-Option*, recorreu-se ao IPERF para gerar 4 fluxos de tráfego (*iperf –c 10.10.10.1 –u –t 120 –i 1 –b 100000*) e ao WireShark para capturar os pacotes. A única diferença passa agora por quebrar a ligação directa entre os Routers 3 e 1. É esperado que o tráfego seja agora reencaminhado para o caminho 2 disponível no túnel, já que o caminho 1 deixa de poder ser utilizado e o caminho 2 é o que apresenta o menor *Path-Option*.

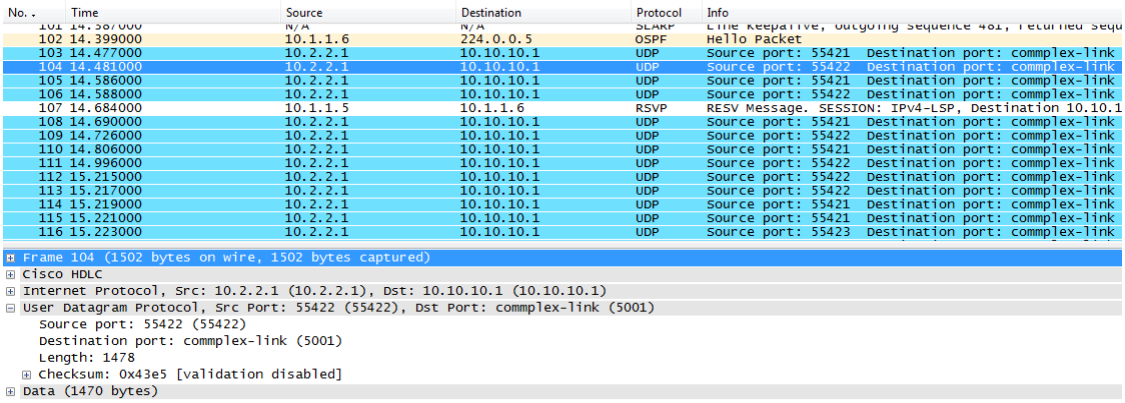


Figura 30 – Captura entre o Router 3 e o Router 1

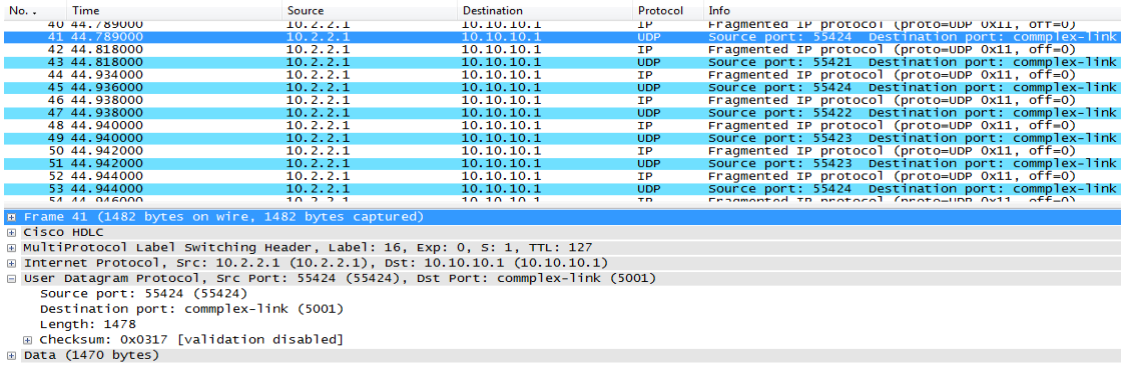


Figura 31 – Captura entre o Router 3 e o Router 4

Mediante a observação da Figura 30 e da Figura 31 verifica-se que o tráfego que passava pelo caminho 1 (Figura 30) passou realmente a ser encaminhado pelo caminho 2 (Figura 31) após a quebra de ligação. Note-se que na Figura 30 apenas se observa o tráfego a passar pelo link que liga o Router 3 ao Router1. Após a quebra de ligação passaram-se a capturar pacotes no link que liga o Router 3 ao Router 4 (Figura 31).

3.7.3. Cenário 3

Para dar continuidade ao estudo, manteve-se o mesmo cenário da subsecção 3.7.1. A grande diferença passa agora pela quebra da ligação entre o Router 3 e o Router 1 e, após um determinado período de tempo, pela quebra da ligação entre o Router 4 e o Router 1. Espera-se que, tal como na secção anterior, após a primeira quebra de ligação o tráfego gerado pelo IPERF passe a circular pelo caminho 2 e após a segunda quebra de ligação passe a circular pelo

caminho 3. Como já havia sido explicado, o tráfego passará em primeiro lugar a circular pelo caminho 2, já que este apresenta um menor *Path-Option*, e só depois da ligação entre estes ser quebrada passará então a circular pelo caminho 3 uma vez que dos três caminhos iniciais este é o que apresenta um valor maior de *Path-Option*.

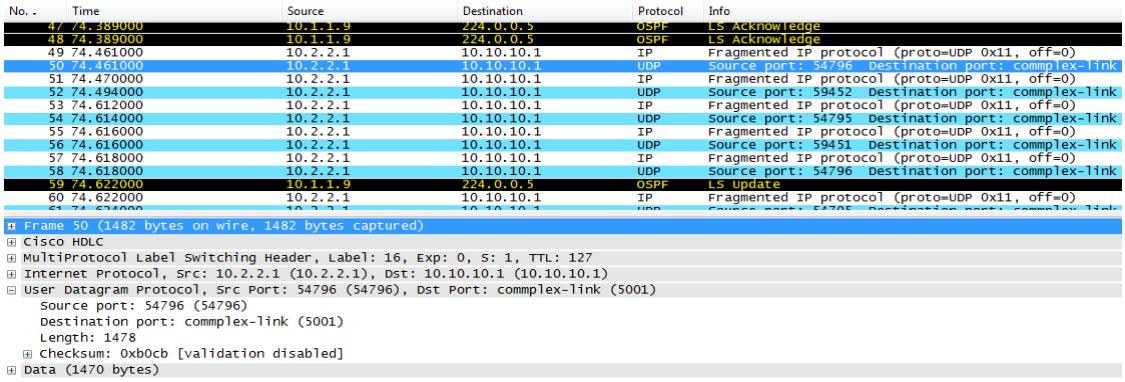


Figura 32 – Captura entre o Router 3 e o Router 2

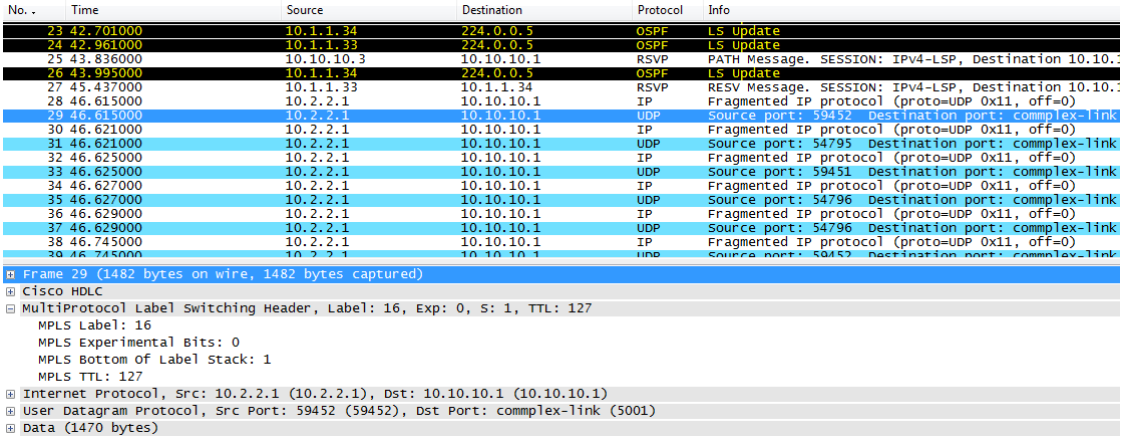


Figura 33 – Captura entre o Router 3 e o Router 4

Através da observação da Figura 32 e Figura 33, pode-se concluir que a nossa experiência foi bem sucedida, uma vez que tal como era esperado após cada quebra de ligação o tráfego passa a transitar para o caminho com o menor Path-Option, desde que este esteja disponível.

3.7.4. Cenário 4

Com o objectivo de se obter uma contra prova de que realmente o sistema muda de um caminho para o outro quando uma das ligações é quebrada e que esta alteração respeita em absoluto o valor do *Path-Option* que foi atribuído ao caminho, alteraram-se os valores dos *Path-Options* correspondentes aos diferentes caminhos. Desta forma, no Router 3 realizou-se a seguinte alteração:

- Path-Option1: caminho 2
- Path-Option3: caminho 3
- Path-Option5: caminho 1

Recorreu-se novamente ao IPERF para gerar 4 fluxos de pacotes (*iperf –c 10.10.10.1 –u –t 120 –i 1 –b 100000*). Pretende-se verificar se o fluxo de informação passa pelo caminho 2, uma vez que este apresenta agora o valor mais baixo de *Path-Option*.

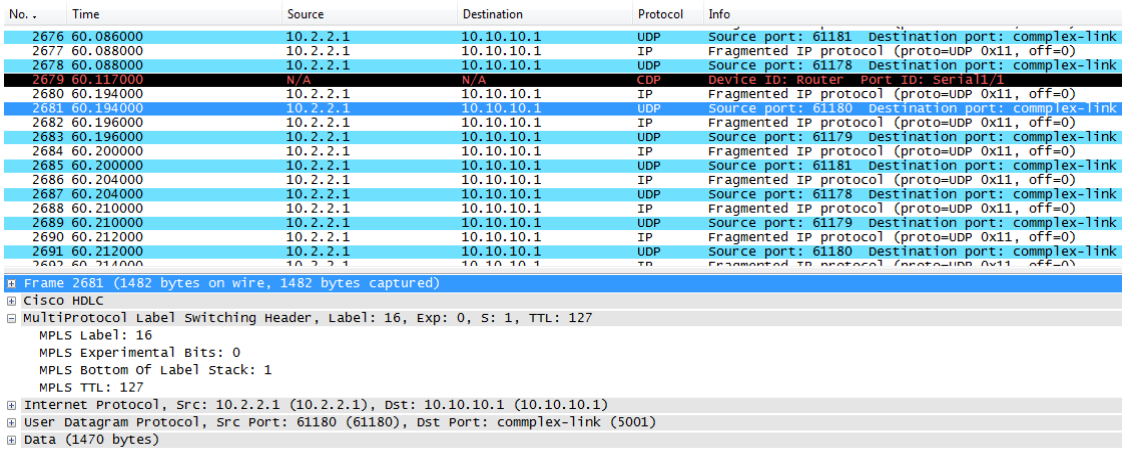


Figura 34 – Captura entre o Router 3 e o Router 4

Mais uma vez os resultados obtidos estão de acordo com o que era esperado, pois como se pode verificar através da Figura 34 os pacotes enviados passam todos apenas pelo caminho 2 (foram realizadas capturas em todos os links mas apenas foram capturados pacotes no link entre o Router 3 e o Router 4, que corresponde ao caminho 2).

3.7.5. Cenário 5

Neste cenário quebraram-se as ligações entre o Router 4 e o Router 1 e posteriormente entre o Router 2 e o Router 1. Geraram-se os mesmos 4 fluxos de pacotes do cenário anterior, mediante a utilização do IPERF. Como tal, é esperado nesta experiência que o fluxo de informação passe primeiro pelo caminho 2, após o primeiro corte de ligação a informação passe pelo caminho 3 e, após o segundo corte, passe pelo caminho 1, respeitando assim os números atribuídos aos *Path-Options*.

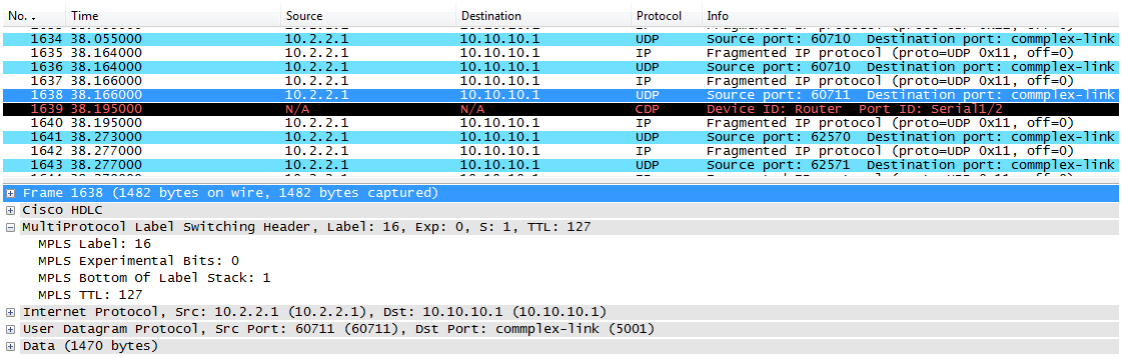


Figura 35 – Captura entre o Router 3 e o Router 4

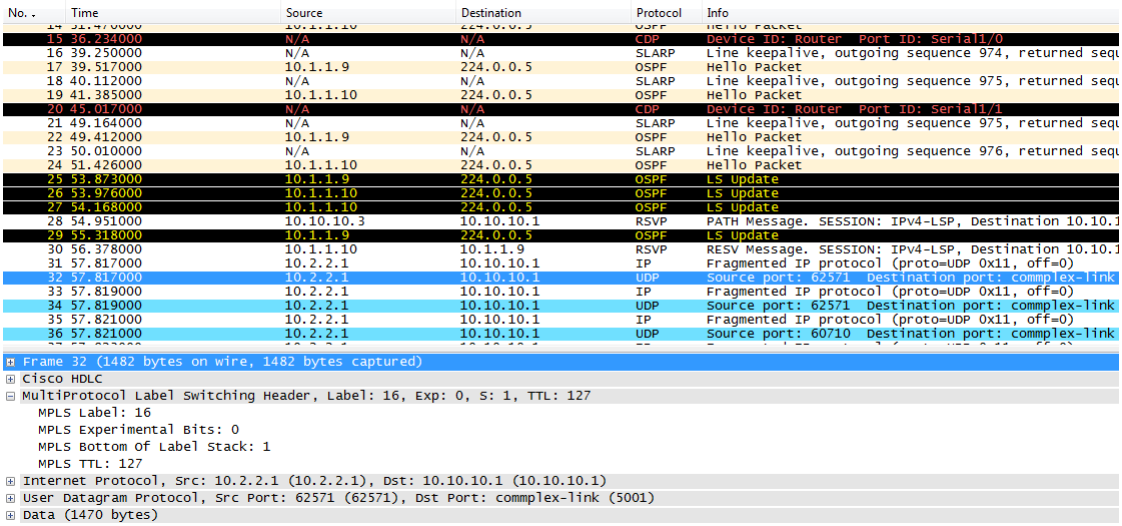


Figura 36 – Captura entre o Router 3 e o Router 2

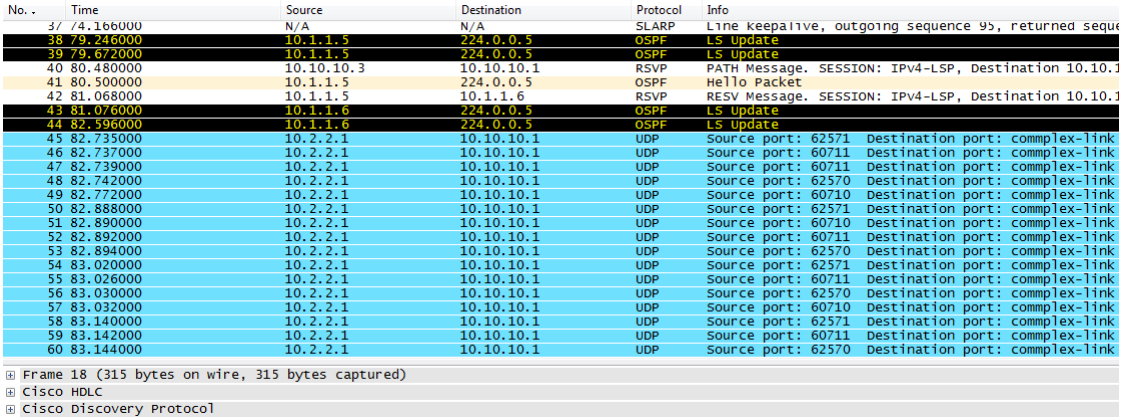


Figura 37 – Captura entre o Router 3 e o Router 1

Na Figura 35 pode-se observar a quebra da primeira ligação, na Figura 36 a quebra da segunda ligação e na Figura 37 a quebra da terceira ligação, passando a informação a ser toda transmitida através do caminho 1, o único disponível.

Em suma, estes dois últimos exercícios servem para comprovar os resultados otbidos em 3.7.2 e 3.7.3, ou seja, os caminhos que vão sendo seleccionados pelos routers dependem realmente do número atribuído no *Path-Option* e não da ordem pela qual os caminhos foram criados ou programados. Verificou-se também que a função do *Path-Option* é garantir sempre a entrega da informação através dos recursos disponíveis: quando de quebrava uma ligação física, a informação passa a ser automaticamente transmitida por outro caminho que esteja disponível.

3.8. Estudo do parâmetro Load Share

O balanceamento de carga entre túneis MPLS TE é realizado no *Headend* entre os túneis que têm os mesmos prefixos de destino. A diferença entre o balanceamento de carga MPLS TE e o fornecido pelo protocolo OSPF é que os túneis MPLS TE permitem realizar o balanceamento de tráfego de maneira proporcional à banda alocada para cada túnel ou de acordo com uma proporção especificada e independente da banda.

O comando utilizado para definir esta proporção é o **tunnel mpls traffic-eng load-share<0-1000000>** . Com a activação deste comando, a proporção de tráfego enviado para cada túnel passa a basear-se no parâmetro configurado no mesmo. Por exemplo, se forem

definidos dois túneis entre R1 e R3, sendo que um tem *load-share* igual a 3 e o outro igual a 1, isto significa que a razão de tráfego entre dois túneis será igual a 1/3.

Para se poder testar o parâmetro *load-share* realizaram-se diversas experiências, tendo como base sempre o mesmo cenário. Este cenário apresenta diversos túneis coom as seguintes características:

- Túnel 1: directamente ligado entre o router 3 e o router 1;
- Túnel 2 : ligado entre o router 3 e o router 1 com passagem pelo router 2.
- Ambos os túneis têm o mesmo *priority* e o mesmo *path-option*.

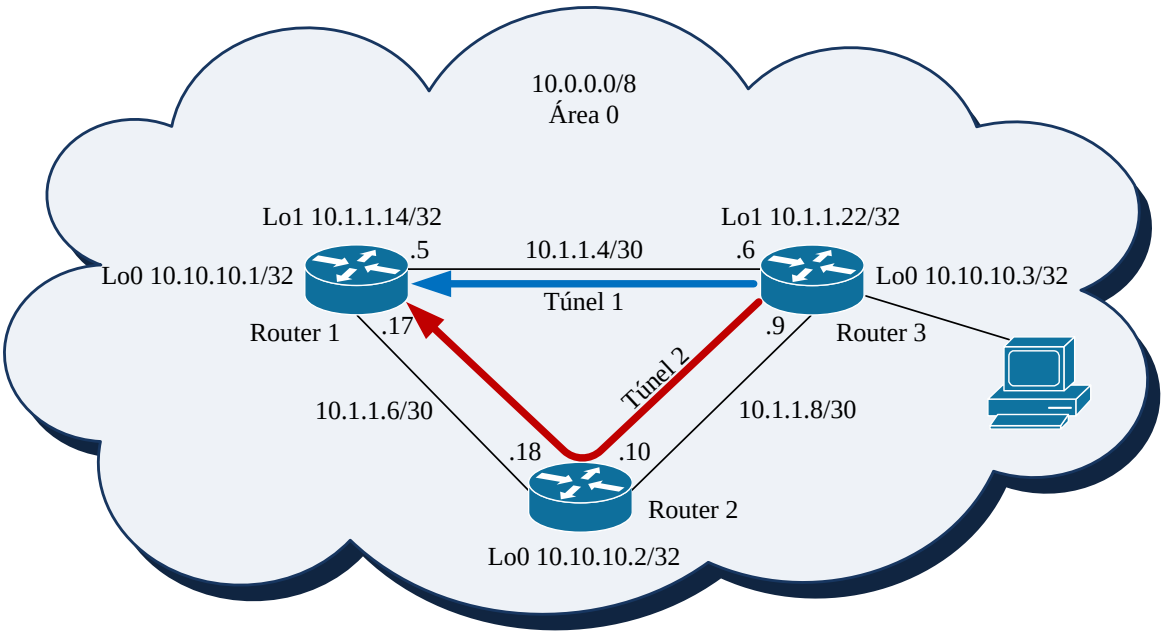


Figura 38 – Cenário para estudo do parâmetro *Load Share*

3.8.1. Cenário 1

Para se verificar o funcionamento da funcionalidade do balanceamento de carga, começou-se por definir dois túneis quase iguais, com a única diferença a residir no valor do *Load-Share*: o Túnel 1 tem um *Load-Share* de 1 e o Túnel 2 um *Load-share* de 3. Tendo em conta que a única diferença nos túneis é o campo *Load-Share*, é de esperar que a carga seja distribuída pelos dois túneis mas em proporções diferentes, ou seja, no túnel 2 a quantidade de carga será superior à que passa no Túnel 1. Para que se possa realizar esta

experiência é necessário recorrer novamente ao IPERF para gerar tráfego suficiente (*iperf –c 10.10.10.1 –u –t 120 –i 1*).

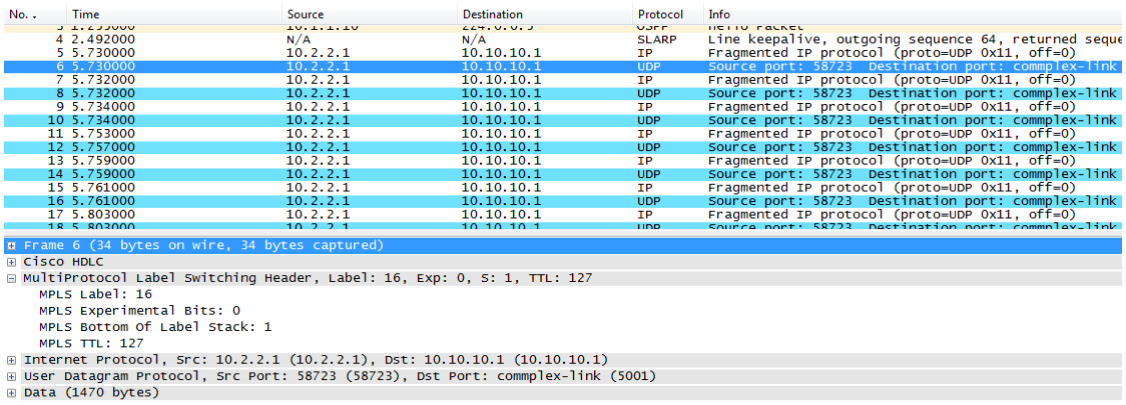


Figura 39 – Captura entre o Router 3 e o Router 2

Mediante a observação destes resultados (Figura 39) verifica-se que todo o tráfego passa pelo túnel 2. Face a este resultado, que vai contra o que era esperado, trocaram-se os valores de *load-share* dos túneis no sentido de verificar se ocorre alguma alteração. Realizou-se então uma nova captura de tráfego.

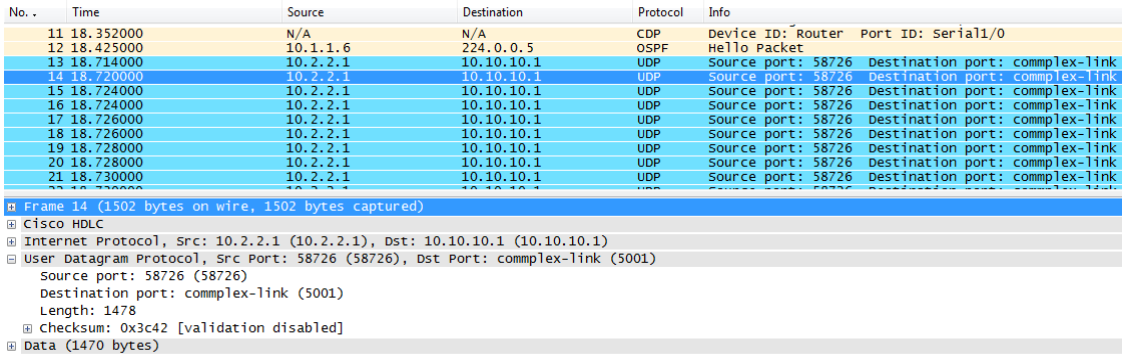


Figura 40 – Captura entre o Router 3 e o Router 1

Verifica-se agora que o tráfego passa todo pelo túnel 1. Esperava-se que o tráfego passasse por ambos os túneis em proporções diferentes, o que não aconteceu, verificando-se apenas que o tráfego passa sempre pelo túnel que tem o valor de *load-share* superior.

3.8.2. Cenário 2

Uma vez que a experiência anterior não correu da forma esperada, nesta experiência mantiveram-se as mesmas definições da experiência anterior mas geraram-se dois fluxos de tráfego (*iperf –c 10.10.10.1 –u –t 120 –i 1*).

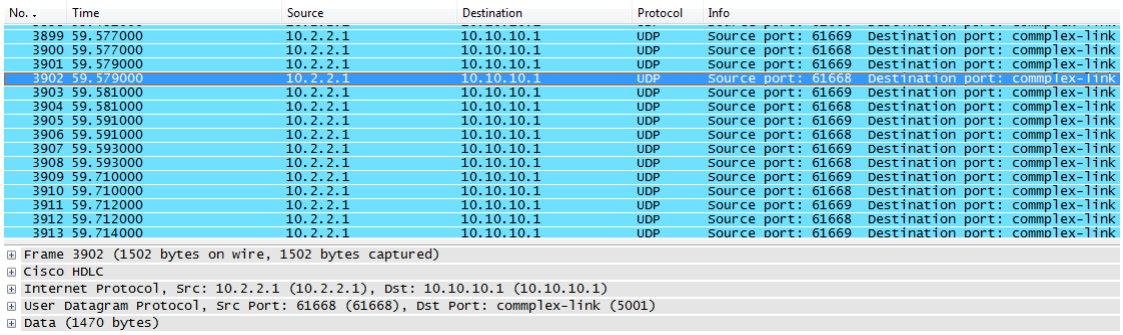


Figura 41 – Captura entre o Router 3 e o Router 1

Da captura de pacotes ilustrada na Figura 41verifica-se que o tráfego gerado continua todo a ser transmitido pelo túnel 1, o que não deveria acontecer, isto é, não existe qualquer balanceamento de carga.

3.8.3. Cenário 3

Uma vez que as experiências anteriores não estão a produzir os resultados esperados, optou-se por testar mais uma hipótese: colocar o load-share com um valor igual em ambos os túneis e gerar (com a ajuda do IPERF) um fluxo de tráfego a partir do terminal que se encontra ligado ao Router 3.

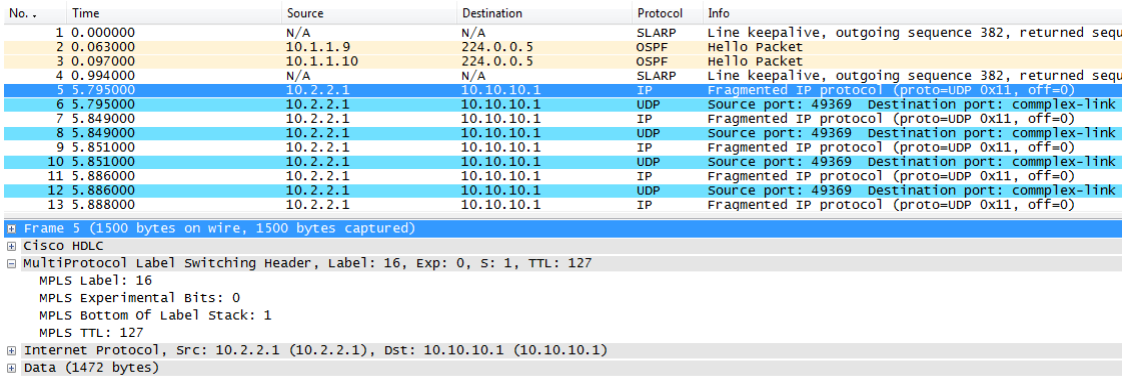


Figura 42 – Captura entre o Router 3 e o Router 2

Verifica-se agora que o fluxo passa todo apenas e só pelo Túnel 2. O problema continua então sem ser solucionado.

3.8.4. Cenário 4

Depois de alguma reflexão sobre o que pode estar a correr menos bem e de se ter revisto todo o procedimento utilizado para testar o *load-share*, o único ponto que continua a suscitar algumas dúvidas é a quantidade de tráfego gerado. O tráfego gerado até este momento é mais do que suficiente para sobrecarregar as ligações e os túneis. Contudo, é estranho que todo o tráfego continue a passar todo num só túnel. Para tirar qualquer dúvida, iremos agora gerar quatro fluxos de tráfego e os túneis apresentam *Load-share* igual a 1 (Túnel 1) e 3 (Túnel 2).

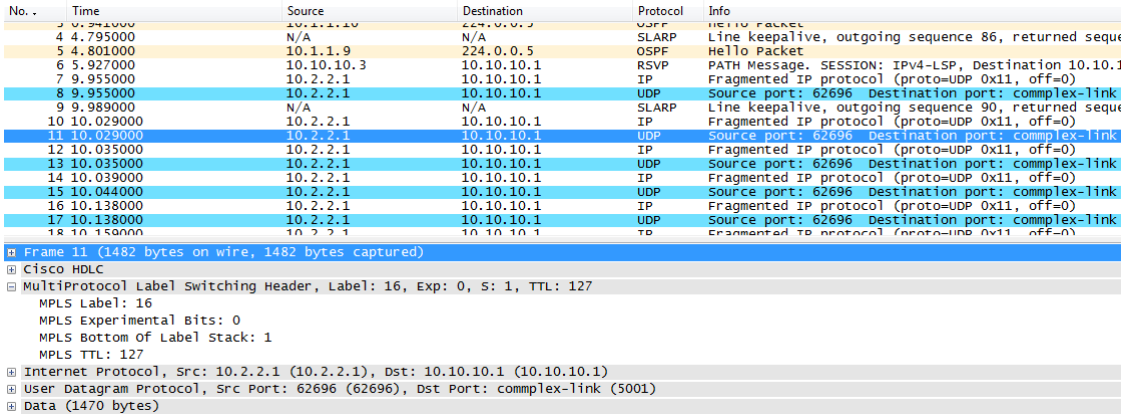


Figura 43 – Captura entre o Router 3 e o Router 2

Da Figura 43 verifica-se mais uma vez que independentemente da quantidade de tráfego gerada os pacotes enviados passam todos pelo Túnel 2, o que começa a suscitar grandes duvidas quanto a fiabilidade do parâmetro *Load-Share*.

No ficheiro da captura, obtido recorrendo a *Statistics->Conversations->UDP*, verificou-se que o tráfego passa realmente todo pelo túnel 2 o que não deveria acontecer até pela limitação da largura de banda das ligações. Como as ligações são de 512K e os túneis têm uma limitação de 200 K, não se entende como é possível tal facto ocorrer. Uma das soluções aponta para o facto de existir algum problema com o simulador ou com o IOS utilizado. Esta última hipótese foi colocada de parte pois já se trocou o IOS dos routers e inclusivamente os próprios modelos de router.

Para descartar a hipótese do erro ser do simulador, recorreu-se ao laboratório para montar a mesma experiência que foi testada no simulador. Os resultados obtidos num cenário real foram exactamente os mesmos que obtivemos no simulador.

4. MPLS e VPN

De modo a estudar a implementação de VPN's em MPLS, montou-se o cenário da Figura 44. Este cenário é composto por dois clientes (A e B) com duas filiais em cidades diferentes, por exemplo. A rede intermédia, que liga e permite a conectividade entre ambas as filiais dos clientes, é uma rede MPLS em que o protocolo OSPF é o IGP. Mais uma vez, foi utilizado o simulador de redes GNS3, exactamente com o mesmo IOS que foi utilizado no Capítulo 3. Os Routers utilizados foram os da série 3700 da CISCO. É também de referir que as ligações na rede MPLS são ligações série com uma largura de banda de 512Kbits/s, sendo utilizado o protocolo HDLC no encapsulamento dos dados. O objectivo destas experiências é não só o de ensinar como se configuram VPNs mas também o de compreender o seu funcionamento, nomeadamente a forma como é feita a distribuição de etiquetas ao longo da rede.

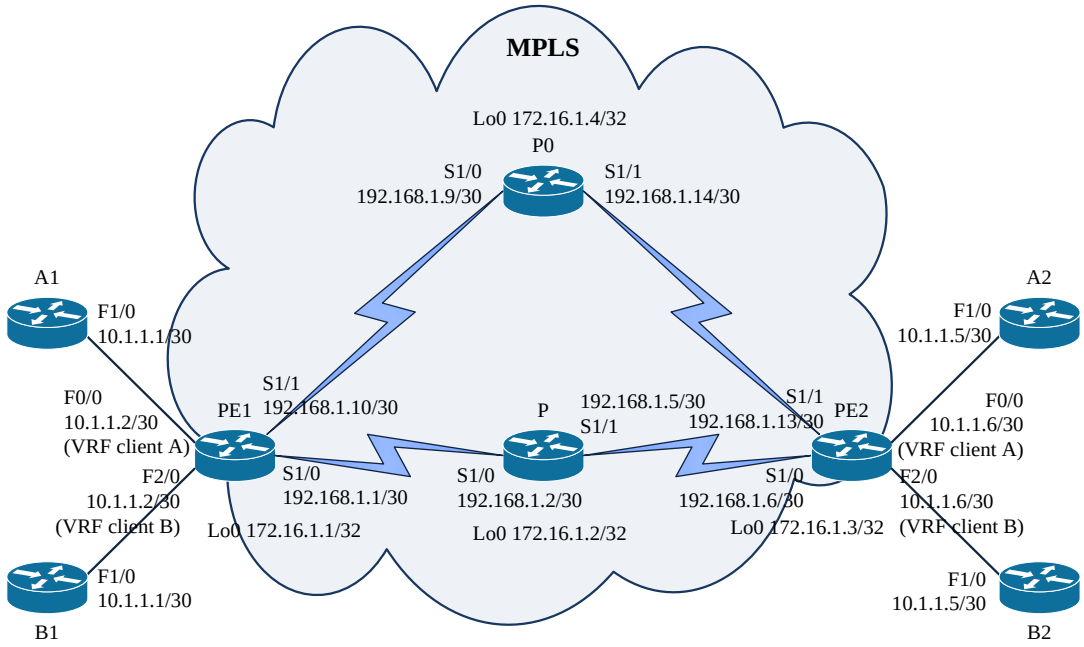


Figura 44 – Cenário para estudo de VPNs com MPLS

4.1. Experiência 1

Na primeira experiência foi apenas feita a programação necessária dos Routers. Configuraram-se as VRFs (*VPN Routing or Forwarding Instances*) para os Clientes A e B. Por exemplo, no Router PE1, introduziram-se os seguintes comandos:

```
PE1(config)#ip vrf ClientA
PE1(config-vrf)#route-target 64999:1                    (VPN ID)
PE1(config-vrf)#rd 999:1                                (route distinguisher)
PE1(config-vrf)#ip vrf ClientB
PE1(config-vrf)#route-target 64999:2
PE1(config-vrf)#rd 999:2
```

O *route distinguisher* permitirá que rotas sobrepostas possam ser distinguidas no backbone MP-BGP.

Depois de se terem definido as VRFs, é-lhes atribuido um interface de modo a que o tráfego que vem encaminhado para essa interface possa utilizar a tabela de VRFs existente:

```
PE1(config-vrf)#int F0/0
PE1(config-if)#ip vrf forwarding ClientA
PE1(config-if)#ip address 10.1.1.2 255.255.255.252
PE1(config-if)#no shut
PE1(config-if)#int F2/0
PE1(config-if)#ip vrf forwarding ClientB
PE1(config-if)#ip address 10.1.1.2 255.255.255.252
PE1(config-if)#no shut
```

Realizou-se uma configuração similar para o Router PE2, tendo apenas em atenção a dirença dos endereços IP, isto é, em vez de 10.1.1.2/30 utilizou-se 10.1.1.6/30.

Agora que as VRFs estão configuradas, tem que existir uma forma de comunicar as rotas contidas nas VRFs a toda a rede. Desta forma, é necessário configurar um protocolo de Routing, neste caso o MP-BGP.
O MP-BGP atribui a capacidade de interligar domínio de routing OSI sem fundir os domínios, o que confere a possibilidade de criar redes OSI bastante vastas. Os benefícios de usar esta capacidade não está restrita a redes DCN (Data Communications Network, rede de comunicação de dados) e pode ser implementado para auxiliar o dimensionamento de qualquer rede que use routing OSI com CLNS (ConnectionLess Network Service).
Assim, no Router PE1 introduziram-se os seguintes comandos:

```
PE1(config)#router bgp 64999
PE1(config-router)#no bgp default ipv4-unicast
PE1(config-router)#neighbor 172.16.1.3 remote-as 64999
PE1(config-router)#neighbor 172.16.1.3 update-source Lo0
PE1(config-router)#address-family vpnv4                 Exchange VPNv4 routes
PE1(config-router-af)#neighbor 172.16.1.3 activate      Activate the neighbour
```

Realizou-se uma configuração semelhante no Router PE2.

Configurou-se agora o Protocolo RIP nos routers A1, B1, A2 e B2, bem como a respectiva redistribuição:

```
A1(config)#router rip
A1(config-router)#version 2
A1(config-router)#network 10.0.0.0
A1(config-router)#no auto-summary
```

No Roter PE1, configurou-se o RIP sobre o VRF:

```
PE1(config)#router rip
PE1(config-router)#version 2
PE1(config-router)#address-family ipv4 vrf ClientA
PE1(config-router-af)#version 2
PE1(config-router-af)#network 10.0.0.0
PE1(config-router-af)#no auto-summary
PE1(config-router)#address-family ipv4 vrf ClientB
PE1(config-router-af)#version 2
PE1(config-router-af)#network 10.0.0.0
PE1(config-router-af)#no auto-summary
```

O RIP está a ser executado entre os PEs e os CEs, enquanto que o MP-BGP se encontra a correr entre os PEs. Então, é necessário redistribuir as rotas entre RIP e BGP e vice-versa:

```
PE1(config)#router bgp 64999
PE1(config-router)#address-family ipv4 vrf ClientA
PE1(config-router-af)#redistribute rip metric 1
PE1(config-router-af)#address-family ipv4 vrf ClientB
PE1(config-router-af)#redistribute rip metric 1
-------------------------------------------------------------------------------------------
PE1(config-router-af)#router rip
PE1(config-router)#address-family ipv4 vrf ClientA
PE1(config-router-af)#redistribute bgp 64999 metric 1
PE1(config-router-af)#address-family ipv4 vrf ClientB
PE1(config-router-af)#redistribute bgp 64999 metric 1
```

Realizou-se uma configuração similar no Router PE2

Após a programação ter sido concluída, os diferentes routers apresentam a seguinte informação:

Router PE1

```
Router#sh mpls ip binding
 172.16.1.1/32
      in label:     imp-null
      out label:    16       lsr: 172.16.1.4:0
      out label:    16       lsr: 172.16.1.2:0
 172.16.1.2/32
      in label:     18
      out label:    17       lsr: 172.16.1.4:0
      out label:    imp-null  lsr: 172.16.1.2:0     inuse
 172.16.1.3/32
      in label:     19
      out label:    18       lsr: 172.16.1.4:0     inuse
      out label:    17       lsr: 172.16.1.2:0     inuse
 172.16.1.4/32
      in label:     16
      out label:    imp-null  lsr: 172.16.1.4:0     inuse
      out label:    18       lsr: 172.16.1.2:0
```

Para a Router 172.16.1.3 (PE2) a etiqueta colocada é a que tem o número de identificação nº 17, pois esta é a que indica a passagem pelo Router P (172.16.1.2).

```
  192.168.1.0/30
      in label:     imp-null
      out label:    19       lsr: 172.16.1.4:0
      out label:    imp-null  lsr: 172.16.1.2:0
  192.168.1.4/30
      in label:     20
      out label:    20       lsr: 172.16.1.4:0
      out label:    imp-null  lsr: 172.16.1.2:0     inuse
  192.168.1.8/30
      in label:     imp-null
      out label:    imp-null  lsr: 172.16.1.4:0
      out label:    19       lsr: 172.16.1.2:0
  192.168.1.12/30
      in label:     17
      out label:    imp-null  lsr: 172.16.1.4:0     inuse
      out label:    20       lsr: 172.16.1.2:0
Router#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     172.16.1.4/32     0          Se1/1      point2point
17     Pop tag     192.168.1.12/30   0          Se1/1      point2point
18     Pop tag     172.16.1.2/32     44         Se1/0      point2point
19     18          172.16.1.3/32     0          Se1/1      point2point
       17          172.16.1.3/32     0          Se1/0      point2point
20     Pop tag     192.168.1.4/30    0          Se1/0      point2point
21     Aggregate   10.1.1.0/30[V]    1040
22     Aggregate   10.1.1.0/30[V]    0
Router#sh ip bgp all
For address family: IPv4 Unicast

For address family: IPv6 Unicast

For address family: VPNv4 Unicast
BGP table version is 9, local router ID is 172.16.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 999:1 (default for vrf ClientA)
*> 10.1.1.0/30      0.0.0.0                  0         32768 ?
*>i10.1.1.4/30      172.16.1.3               0    100      0 ?
Route Distinguisher: 999:2 (default for vrf ClientB)
*> 10.1.1.0/30      0.0.0.0                  0         32768 ?
*>i10.1.1.4/30      172.16.1.3               0    100      0 ?

For address family: IPv4 Multicast

For address family: IPv6 Multicast

For address family: NSAP Unicast
Router#sh ip route vrf ClientA

Routing Table: ClientA
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/30 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
B       10.1.1.4 [200/0] via 172.16.1.3, 00:19:49
Router#sh ip route vrf ClientB

Routing Table: ClientB
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/30 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, FastEthernet2/0
B       10.1.1.4 [200/0] via 172.16.1.3, 00:20:07
```

Router PE2

```
Router#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     16          172.16.1.1/32    0          Se1/1       point2point
       16          172.16.1.1/32    0          Se1/0       point2point
17     Pop tag     172.16.1.2/32    0          Se1/0       point2point
18     Pop tag     172.16.1.4/32    0          Se1/1       point2point
19     Pop tag     192.168.1.0/30   0          Se1/0       point2point
20     Pop tag     192.168.1.8/30   0          Se1/1       point2point
21     Aggregate   10.1.1.4/30[V]   1040
22     Aggregate   10.1.1.4/30[V]   0
Router#sh mpls ip binding
  172.16.1.1/32
      in label:   16
      out label:  16      lsr: 172.16.1.4:0    inuse
      out label:  16      lsr: 172.16.1.2:0    inuse
  172.16.1.2/32
      in label:   17
      out label:  17      lsr: 172.16.1.4:0
      out label:  imp-null lsr: 172.16.1.2:0   inuse
  172.16.1.3/32
      in label:   imp-null
      out label:  18      lsr: 172.16.1.4:0
      out label:  17      lsr: 172.16.1.2:0
  172.16.1.4/32
      in label:   18
      out label:  imp-null lsr: 172.16.1.4:0   inuse
      out label:  18      lsr: 172.16.1.2:0
  192.168.1.0/30
```

Para a Router 172.16.1.1 (PE1) a etiqueta colocada é a que tem o número de identificação nº 16, pois esta é a que indica a passagem pelo Router P0 (172.16.1.4).

```
      in label:     19
      out label:    19       lsr: 172.16.1.4:0
      out label:    imp-null  lsr: 172.16.1.2:0     inuse
  192.168.1.4/30
      in label:     imp-null
      out label:    20       lsr: 172.16.1.4:0
      out label:    imp-null  lsr: 172.16.1.2:0
  192.168.1.8/30
      in label:     20
      out label:    imp-null  lsr: 172.16.1.4:0     inuse
      out label:    19       lsr: 172.16.1.2:0
  192.168.1.12/30
      in label:     imp-null
      out label:    imp-null  lsr: 172.16.1.4:0
      out label:    20       lsr: 172.16.1.2:0
Router#sh ip bgp all
For address family: IPv4 Unicast

For address family: IPv6 Unicast

For address family: VPNv4 Unicast
BGP table version is 9, local router ID is 172.16.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 999:1 (default for vrf ClientA)
*>i10.1.1.0/30     172.16.1.1               0    100      0 ?
*> 10.1.1.4/30     0.0.0.0                  0         32768 ?
Route Distinguisher: 999:2 (default for vrf ClientB)
*>i10.1.1.0/30     172.16.1.1               0    100      0 ?
*> 10.1.1.4/30     0.0.0.0                  0         32768 ?

For address family: IPv4 Multicast

For address family: IPv6 Multicast

For address family: NSAP Unicast
Router#sh ip route vrf ClientA

Routing Table: ClientA
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 2 subnets
B       10.1.1.0 [200/0] via 172.16.1.1, 00:25:14
C       10.1.1.4 is directly connected, FastEthernet0/0
Router#sh ip route vrf ClientB

Routing Table: ClientB
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 2 subnets
B       10.1.1.0 [200/0] via 172.16.1.1, 00:25:37
C       10.1.1.4 is directly connected, FastEthernet2/0
```

Router P:

```
Router#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     172.16.1.1/32     0          Se1/0       point2point
17     Pop tag     172.16.1.3/32     1080       Se1/1       point2point
18     18          172.16.1.4/32     0          Se1/1       point2point
       16          172.16.1.4/32     0          Se1/0       point2point
19     Pop tag     192.168.1.8/30    0          Se1/0       point2point
20     Pop tag     192.168.1.12/30   0          Se1/1       point2point
Router#sh mpls ip binding
  172.16.1.1/32
      in label:    16
      out label:   imp-null  lsr: 172.16.1.1:0     inuse
      out label:   16        lsr: 172.16.1.3:0
  172.16.1.2/32
      in label:    imp-null
      out label:   18        lsr: 172.16.1.1:0
      out label:   17        lsr: 172.16.1.3:0
  172.16.1.3/32
      in label:    17
      out label:   19        lsr: 172.16.1.1:0
      out label:   imp-null  lsr: 172.16.1.3:0     inuse
  172.16.1.4/32
      in label:    18
      out label:   16        lsr: 172.16.1.1:0     inuse
      out label:   18        lsr: 172.16.1.3:0     inuse
  192.168.1.0/30
      in label:    imp-null
      out label:   imp-null  lsr: 172.16.1.1:0
      out label:   19        lsr: 172.16.1.3:0
  192.168.1.4/30
      in label:    imp-null
      out label:   20        lsr: 172.16.1.1:0
      out label:   imp-null  lsr: 172.16.1.3:0
  192.168.1.8/30
      in label:    19
      out label:   imp-null  lsr: 172.16.1.1:0     inuse
      out label:   20        lsr: 172.16.1.3:0
  192.168.1.12/30
      in label:    20
      out label:   17        lsr: 172.16.1.1:0
```

Pode-se observar que a etiqueta tem como destino o Router PE2(172.16.1.3), tal como já havia sido visto vem identificada com o número 17 (*in label*).

```
    out label:    imp-null  lsr: 172.16.1.3:0     inuse
```

Router P0:

```
Router#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     172.16.1.1/32    5112       Se1/0       point2point
17     17          172.16.1.2/32    0          Se1/1       point2point
       18          172.16.1.2/32    48         Se1/0       point2point
18     Pop tag     172.16.1.3/32    4120       Se1/1       point2point
19     Pop tag     192.168.1.0/30   0          Se1/0       point2point
20     Pop tag     192.168.1.4/30   0          Se1/1       point2point
Router#sh mpls ip binding
 172.16.1.1/32
      in label:    16
      out label:   imp-null  lsr: 172.16.1.1:0     inuse
      out label:   16        lsr: 172.16.1.3:0
 172.16.1.2/32
      in label:    17
      out label:   17        lsr: 172.16.1.3:0     inuse
      out label:   18        lsr: 172.16.1.1:0     inuse
 172.16.1.3/32
      in label:    18
      out label:   imp-null  lsr: 172.16.1.3:0     inuse
      out label:   19        lsr: 172.16.1.1:0
 172.16.1.4/32
      in label:    imp-null
      out label:   16        lsr: 172.16.1.1:0
      out label:   18        lsr: 172.16.1.3:0
 192.168.1.0/30
      in label:    19
      out label:   imp-null  lsr: 172.16.1.1:0     inuse
      out label:   19        lsr: 172.16.1.3:0
 192.168.1.4/30
      in label:    20
      out label:   imp-null  lsr: 172.16.1.3:0     inuse
      out label:   20        lsr: 172.16.1.1:0
 192.168.1.8/30
      in label:    imp-null
      out label:   imp-null  lsr: 172.16.1.1:0
      out label:   20        lsr: 172.16.1.3:0
 192.168.1.12/30
      in label:    imp-null
      out label:   17        lsr: 172.16.1.1:0
      out label:   imp-null  lsr: 172.16.1.3:
```

Pode-se observar que a etiqueta tem como destino o Router PE1(172.16.1.1), tal como já havia sido visto vem identificada com o número 16 (*in label*).

Router A2:

```
Router#sh mpls forwarding-table
Tag switching is not operational.
CEF or tag switching has not been enabled.
No TFIB currently allocated.
Router#sh mpls ip binding
LIB not enabled
```

```
Router#sh mpls ip bgp neighbors
                ^
% Invalid input detected at '^' marker.

Router#sh ip bgp all
% BGP not active

Router#sh ip route vrf ClientA
% IP routing table ClientA does not exist
```

Após se analisar a informação contida nos routers, principalmente a informação relativa às etiquetas, efectuou-se um ping do Router A1 para o Router A2. Este tráfego vai utilizar a VRF ClientA. Para tal, vão ser colocadas etiquetas MPLS no *Ingress Router*, neste caso o Router PE1. São colocadas duas etiquetas, tal como se pode observar na captura realizada (Figura 45 e Figura 46): uma etiqueta com o nº 21 e outra com o nº 17. A etiqueta com o nº 21 é adicionada pelo ingress Router (PE1), pois estes pacotes são originados na VPN. O label 17 é adicionado no topo da pilha de labels porque a LIB (Label Information Base) do Router PE1 tem uma entrada, recebida via LDP do 172.16.1.2 (router P), o seu next-hop para a rede de destino, que obriga à adição deste label aos pacotes com destino ao *Egress Router* (Router PE"), do ponto de vista da rede MPLS. Entre os Routers P e PE2, o label 17 é retirado da pilha de labels pelo router P, expondo o label 21. Este comportamento deriva da LIB de P, que indica que ao chegar um pacote com o label 17 no topo da pilha, com destino ao egress Router PE2 este deva ser retirado, expondo o que está imediatamente a seguir, neste caso o label 21. O Router PE retira ainda o último label, encaminhando os pacotes *ICMP Echo Request* para a rede 10.1.1.0/30.

Os pacotes *ICMP Echo Reply* entre os routers PE2 e P0 têm dois cabeçalhos MPLS, um com o label 16 e outro com o label 21, adicionados pelo o router PE2. O label 21 é adicionado porque a origem destes pacotes está na VPN. O label 16 é adicionado em cima do label 21, porque existe uma entrada LIB do PE2, recebida por LDP do router P0 (172.16.1.4), o next-hop para a rede 10.1.1.0/30, que indica que os pacotes com destino ao Egress Router PE1 (172.16.1.1), do ponto de vista da rede MPLS, devem conter labels (out label: 16). Entre os Routers P0 e PE1, estes pacotes apenas contém o cabeçalho MPLS com label 21 (referente à VPN), porque o Router P0 retirou o label 16 do topo da pilha nos pacotes recebidos por PE2, expondo o label 21. A LIB do router P0 tem uma entrada, recebida por LDP do PE1, que indica que nos pacotes com destino ao Egress Router PE1,

do ponto de vista da rede MPLS, deve ser retirado o label 16. Finalmente o Router PE1, encaminha os pacotes *ICMP Echo Reply* para a rede 10.1.1.0/30, retirando-lhes o label 21.

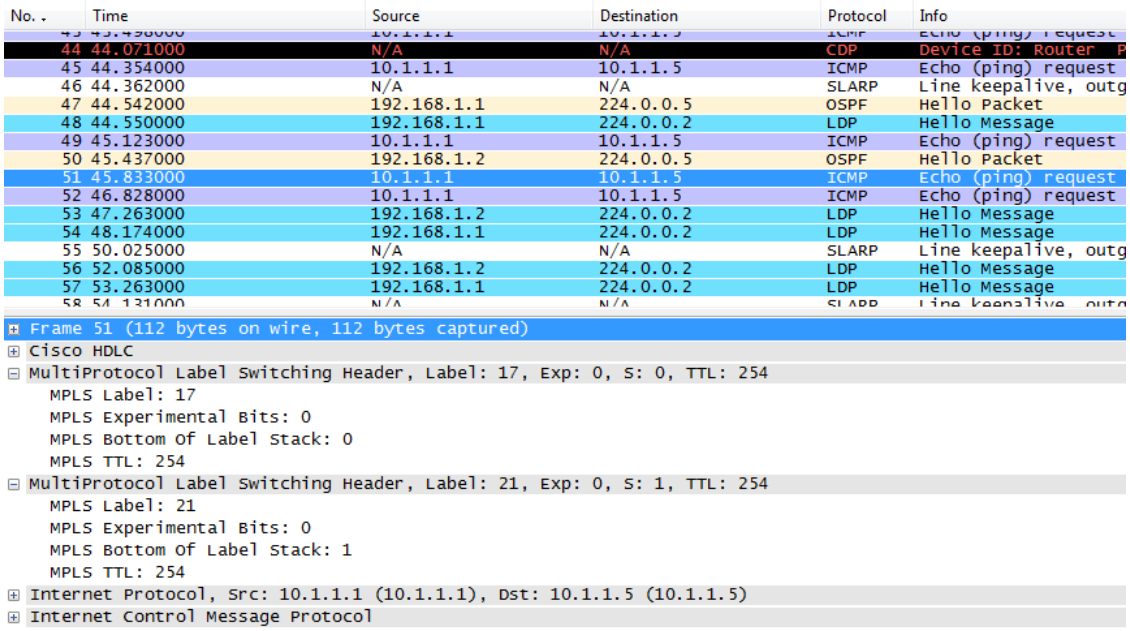


Figura 45 – Captura PE1_P

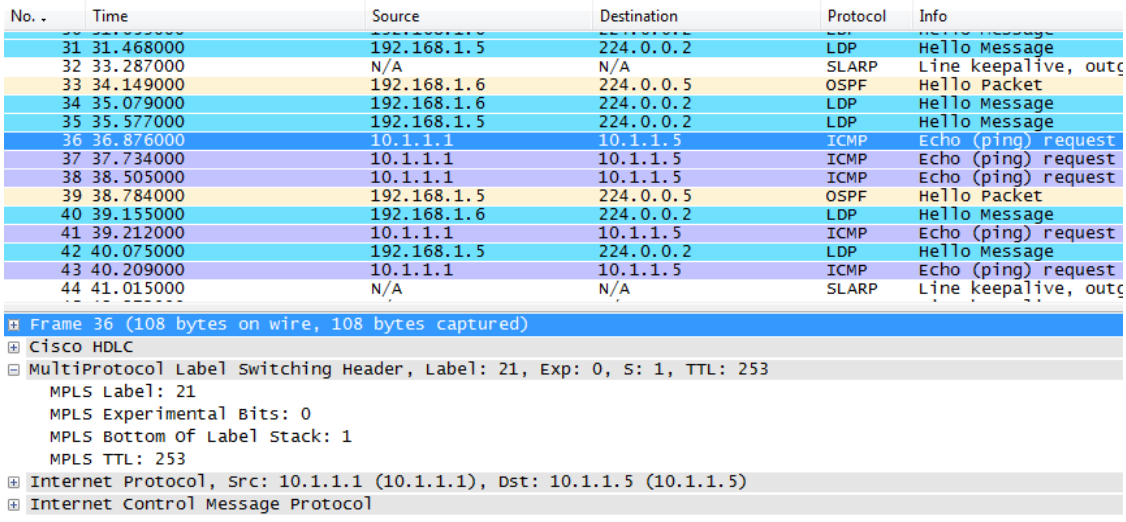


Figura 46 – Captura PE2_P

4.2. Experiência 2

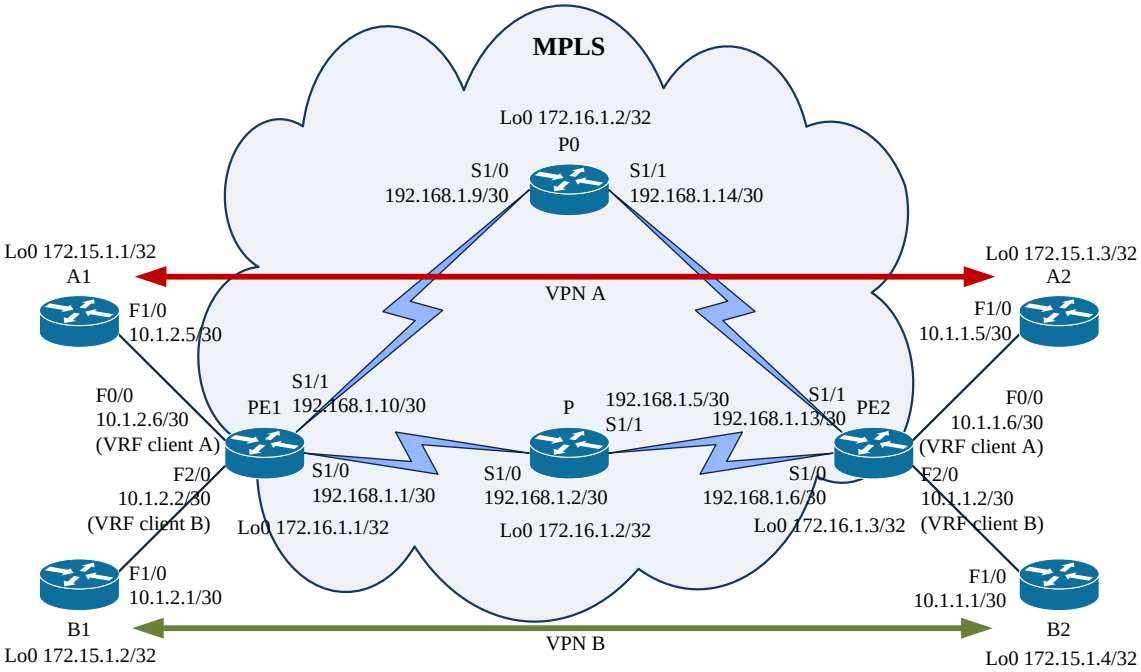


Figura 47 – Estudo de VPNs:Representação das VPNs dos Clientes

Através da observação dos resultados da experiência anterior, verifica-se que os pacotes enviados e recebidos percorrem caminhos diferentes. De seguida, procurou-se perceber porquê. Alteraram-se os custos das portas num dos Routers (P0) por forma a que qualquer caminho que envolva a passagem por este Router apresente um custo superior: assim, alterou-se o custo das portas do Router P0 de 64 para 100. Efectuando ping do Router A1 para A2 e realizando algumas capturas consegue perceber-se se existe algum tipo de pacotes que passe por algum caminho que envolva P0.

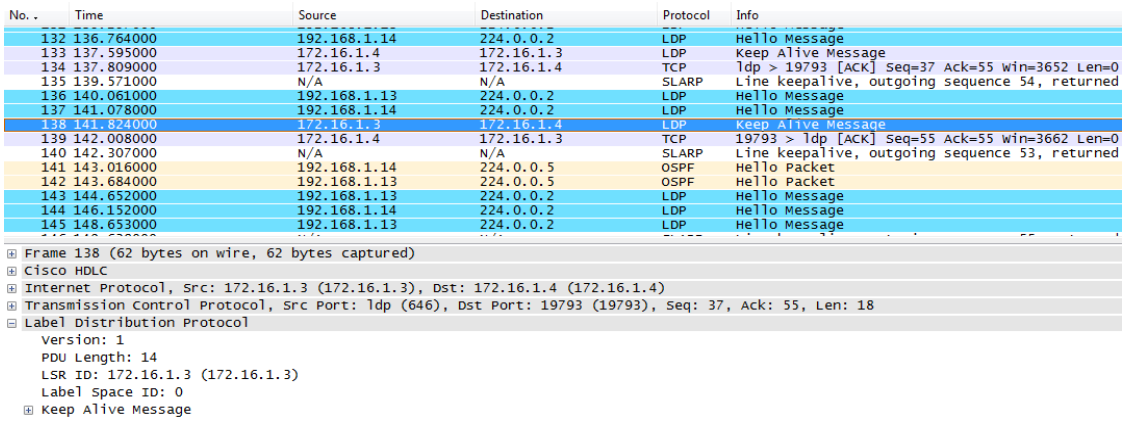
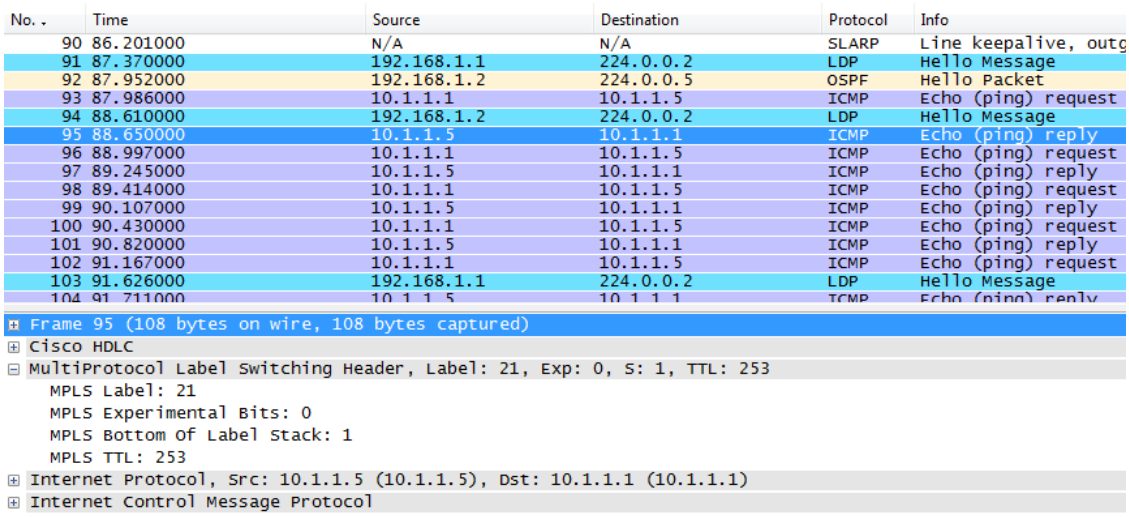


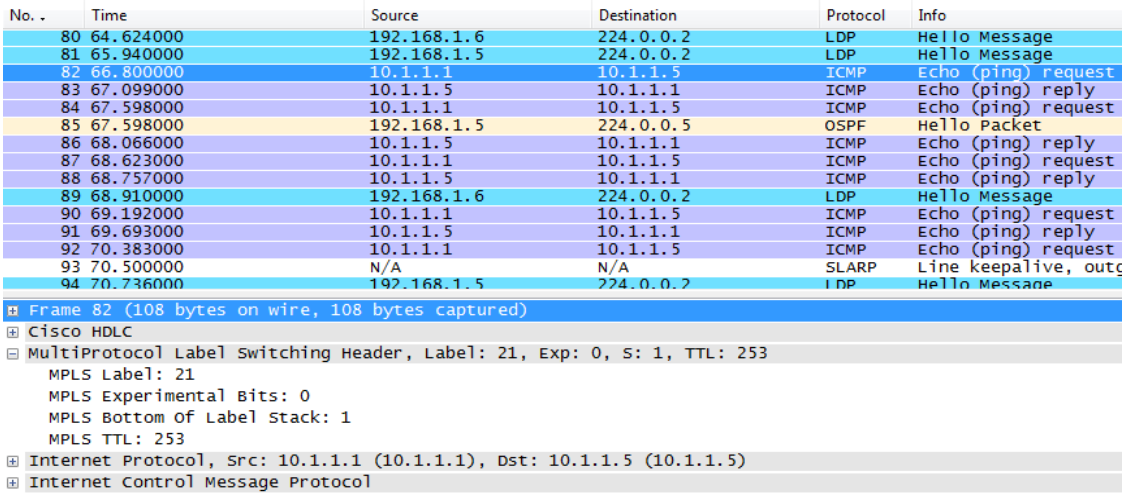
Figura 48 – Captura PE2_P0_cost



No.	Time	Source	Destination	Protocol	Info
90	86.201000	N/A	N/A	SLARP	Line keepalive, outg
91	87.370000	192.168.1.1	224.0.0.2	LDP	Hello Message
92	87.952000	192.168.1.2	224.0.0.5	OSPF	Hello Packet
93	87.986000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
94	88.610000	192.168.1.2	224.0.0.2	LDP	Hello Message
95	88.650000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
96	88.997000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
97	89.245000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
98	89.414000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
99	90.107000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
100	90.430000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
101	90.820000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
102	91.167000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
103	91.626000	192.168.1.1	224.0.0.2	LDP	Hello Message
104	91.711000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply

```
Frame 95 (108 bytes on wire, 108 bytes captured)
Cisco HDLC
MultiProtocol Label Switching Header, Label: 21, Exp: 0, S: 1, TTL: 253
    MPLS Label: 21
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 253
Internet Protocol, Src: 10.1.1.5 (10.1.1.5), Dst: 10.1.1.1 (10.1.1.1)
Internet Control Message Protocol
```

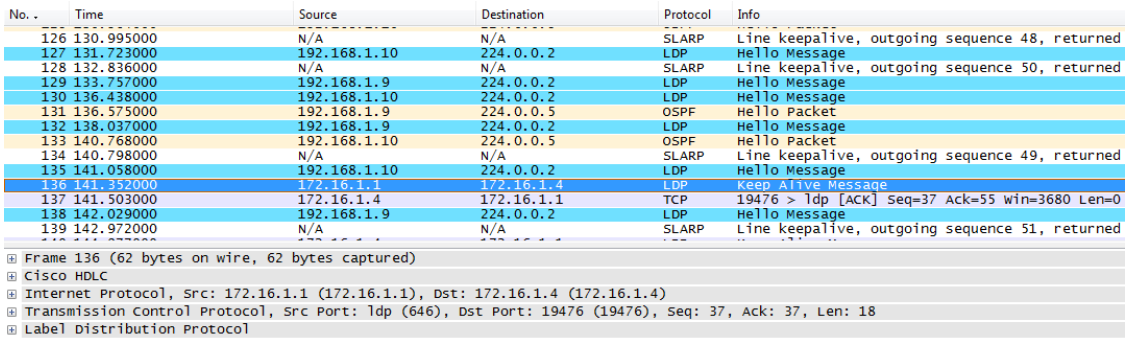
Figura 49 – Captura PE1_P_cost



No.	Time	Source	Destination	Protocol	Info
80	64.624000	192.168.1.6	224.0.0.2	LDP	Hello Message
81	65.940000	192.168.1.5	224.0.0.2	LDP	Hello Message
82	66.800000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
83	67.099000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
84	67.598000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
85	67.598000	192.168.1.5	224.0.0.5	OSPF	Hello Packet
86	68.066000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
87	68.623000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
88	68.757000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
89	68.910000	192.168.1.6	224.0.0.2	LDP	Hello Message
90	69.192000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
91	69.693000	10.1.1.5	10.1.1.1	ICMP	Echo (ping) reply
92	70.383000	10.1.1.1	10.1.1.5	ICMP	Echo (ping) request
93	70.500000	N/A	N/A	SLARP	Line keepalive, outg
94	70.736000	192.168.1.5	224.0.0.2	LDP	Hello Message

```
Frame 82 (108 bytes on wire, 108 bytes captured)
Cisco HDLC
MultiProtocol Label Switching Header, Label: 21, Exp: 0, S: 1, TTL: 253
    MPLS Label: 21
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 253
Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.5 (10.1.1.5)
Internet Control Message Protocol
```

Figura 50 – Captura PE2_P_cost



No.	Time	Source	Destination	Protocol	Info
126	130.995000	N/A	N/A	SLARP	Line keepalive, outgoing sequence 48, returned
127	131.723000	192.168.1.10	224.0.0.2	LDP	Hello Message
128	132.836000	N/A	N/A	SLARP	Line keepalive, outgoing sequence 50, returned
129	133.757000	192.168.1.9	224.0.0.2	LDP	Hello Message
130	136.438000	192.168.1.10	224.0.0.2	LDP	Hello Message
131	136.575000	192.168.1.9	224.0.0.5	OSPF	Hello Packet
132	138.037000	192.168.1.9	224.0.0.2	LDP	Hello Message
133	140.768000	192.168.1.10	224.0.0.5	OSPF	Hello Packet
134	140.798000	N/A	N/A	SLARP	Line keepalive, outgoing sequence 49, returned
135	141.058000	192.168.1.10	224.0.0.2	LDP	Hello Message
136	141.352000	172.16.1.1	172.16.1.4	LDP	Keep Alive Message
137	141.503000	172.16.1.4	172.16.1.1	TCP	19476 > ldp [ACK] Seq=37 Ack=55 Win=3680 Len=0
138	142.029000	192.168.1.9	224.0.0.2	LDP	Hello Message
139	142.972000	N/A	N/A	SLARP	Line keepalive, outgoing sequence 51, returned

```
Frame 136 (62 bytes on wire, 62 bytes captured)
Cisco HDLC
Internet Protocol, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.4 (172.16.1.4)
Transmission Control Protocol, Src Port: ldp (646), Dst Port: 19476 (19476), Seq: 37, Ack: 37, Len: 18
Label Distribution Protocol
```

Figura 51 – Captura PE1_P0_cost

Da observação das capturas realizadas (Figura 48, Figura 49, Figura 50 e Figura 51) verifica-se que após a alteração dos custos das portas do Router P0 mais nenhum pacote enviado passou por este Router, já que os percursos que por ele passam apresentam custos superiores. De facto, verifica-se que as capturas efectuadas nas ligações entre os Routers PE1 e P0 e os Routers PE2 e P0 não incluem qualquer pacote do tipo ICMP. Pelo contrário, as capturas realizadas nas ligações entre os Routers PE1 e PE2 com o Router P possuem os pacotes do tipo ICMP.

Conclui-se assim que é possível fazer o envio e a recepção de informação por caminhos diferentes, uma vez que são os custos associados ao protocolo IGP que determinam os percursos que serão usados pelo MP-BGP. Na experiência apresentada, é utilizado na rede MPLS o protocolo de encaminhamento OSPF, com o mesmo custo em todas as portas. Assim, como o número de interfaces percorridas pelas mensagens é o mesmo, independentemente do caminho tomado, o encaminhamento é feito por qualquer dos caminhos disponíveis.

5. Conclusões

A tecnologia MPLS desempenha um papel cada vez mais importante no contexto das redes IP. Com este trabalho pretendeu-se idealizar um conjunto de experiências de carácter pedagógico capaz de ilustrar os principais conceitos e funcionalidades da tecnologia MPLS. Pretende-se ainda que as experiências definidas sejam utilizadas nas aulas laboratoriais das disciplinas de redes do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e, nesse sentido, todos os cenários idealizados têm em atenção os constrangimentos de equipamento que essas aulas impõem.

Através da realização deste trabalho foi possível sedimentar algumas ideias já adquiridas acerca do MPLS. De facto, o MPLS melhora a eficiência das redes em que é utilizado. Por outro lado, a utilização do MPLS numa rede, nomeadamente a sua configuração nos routers, não é propriamente um processo muito simples: assim, é nossa convicção que este trabalho permite que os alunos que o venham a consultar prestem especial atenção a alguns pormenores, como a largura de banda utilizada ou o tipo de ligações físicas que devem escolher na interligação dos routers.

Uma das funcionalidades mais importantes do MPLS é a possibilidade de implementar engenharia de tráfego baseada em túneis. Existem diversos parâmetros que caracterizam os túneis MPLS, tais como o *Priority, Path-Option* e o *Load-Share*, entre outros. Neste trabalho exploraram-se alguns destes parâmetros no sentido de estudar e ilustrar a sua funcionalidade.

Relativamente ao parâmetro *Priority* fizeram-se algumas descobertas interessantes: por exemplo, quando se criam dois túneis entre os mesmos dois pontos e eles possuem o mesmo caminho, a informação transmitida deveria passar pelo túnel que apresenta uma maior prioridade, mas na realidade, e segundo os resultados obtidos nas experiências realizadas, isto só acontece se as ligações físicas existentes entre os routers impuserem algum tipo de restrições no que diz respeito à largura de banda. Caso contrário, se existir largura de banda suficiente os routers podem optar por estabelecer ambos os túneis e enviar a informação pelos dois. Nas nossas experiências, quando se utilizaram ligações Ethernet para interligar os routers, e uma vez que esta ligações possuem uma largura de banda bastante elevada (10Mbps), eram estabelecidos sempre todos os túneis que foram criados, o que inviabilizou o estudo não só do parâmetro *Prority* mas também de outras

características. Assim, a solução encontrada passou por trocar as ligações físicas existentes por ligações série, que permitem controlar a largura de banda. Assim, o parâmetro *Priority* passou a apresentar a funcionalidade esperada, isto é, o túnel que tiver sido programado com o menor valor no campo *Priority* será o mais prioritário, ou seja, se existirem diversos túneis à escolha, este será o primeiro a ser utilizado.

O *Path-Option* foi o próximo parâmetro a ser estudado. Tal como o parâmetro anterior, também o *Path-Option* apresenta o mesmo comprtamento relativamente à largura de banda disponível nas ligações físicas entre routers. Para além disso, o parâmetro *Priority* é sempre mais importante do que o parâmetro *Path-Option*: independentemente do valor do *Path-Option*, o túnel escolhido será sempre o de maior prioridade. Se tivermos dois túneis com a mesma prioridade e cada um dos túneis possuir valores de *Path-Option* diferentes, a informação pode ser eventualmente distribuída por ambos os Túneis: neste caso, a decisão de escolha do caminho por onde os pacotes serão enviados baseia-se nos protocolos de encaminhamento.

Numa outra situação estudada, tendo agora apenas um túnel e dois caminhos definidos dentro do mesmo túnel que diferem não só no percurso escolhido como também no número associado ao *Path-Option* (quanto menor for o número, maior a prioridade deste caminho, dentro do mesmo túnel), verifica-se que neste caso o fluxo de pacotes vai transitar pelo caminho com menor *Path-Option*. Deve-se, desde já, concluir que o *Path-Option* só é utilizado pelos routers quando existe apenas um túnel definido e vários caminhos disponíveis para enviarem a informação até ao destino. Outra das utilidades do *Path-Option* é visível quando durante a transmissão de informação ocorre uma quebra numa das ligações físicas que faz parte do caminho que está a ser utilizado naquele momento. Nesse caso, os routers, de forma quase imediata, passam a transmitir a informação por outro caminho que esteja disponível, isto é, recorrem ao caminho com o número de *Path-Option* imediatamente a seguir ao que estava a ser utilizado.

Durante o nosso estudo, era também nossa intenção verificar se sobrecarregando um determinado caminho, o router passaria a fazer distribuição de carga. No entanto, a experiência não foi bem sucedida, já que de forma inexplicável o tráfego era sempre transmitido apenas por um caminho.

Em suma, na transmissão de informação os routers atendem primeiro à prioridade dos túneis. Dentro dos túneis atendem primeiro ao caminho que tiver o menor *Path-*

Option. Os caminhos alternativos disponíveis são utilizados na maior parte das vezes como caminhos de *backup*, isto é, em caso de falha os routers utilizam o caminho seguinte no sentido de diminuir os tempos de espera e as perdas de informação.

Estudados os parâmetros *Path-Option* e *Priority*, o estudo avançou para o parâmetro *Load-Share*. Como o próprio nome indica, este campo permite que se efectue um balanceamento de carga nos túneis. Isto permite que os túneis não fiquem sobrecarregados de tráfego. O que acontece é que durante o estudo efectuado tal facto nunca se verificou, por mais testes que fossem efectuados, quer no simulador quer em ambiente real. Por incrível que pareça, por maior que fosse a quantidade de tráfego gerado, este passava sempre pelo mesmo túnel: nem a imposição de ligações série limitava o tráfego. Após muita pesquisa, efectuada já durante a escrita desta dissertação, foi finalmente descoberta uma justificação para este facto. Apenas se forem programadas instruções de qualidade de serviço (QoS) nos routers é que o balanceamento de carga funcionará, pois sem mecanismos de QoS a correr nos routers o balanceamento de carga é pura e simplesmente ignorado por estes. Dada a resolução tardia deste problema, já não houve oportunidade de testar se realmente o balanceamento de carga funcionaria na perfeição quando se utilizam mecanismos de QoS.

Por fim, foram estudadas VPNs sobre MPLS, dada a sua importância actual no contexto das redes IP. Nestas experiências utilizou-se um número maior de routers de modo a criarem-se cenários mais próximos da realidade. Concretamente, foi criado um cenário correspondente a duas empresas, com duas sucursais cada e localizadas em cidades diferentes. Ambas as empresas necessitam de realizar troca de informação entre as sucursais e não desejam que a informação seja acedida por entidades externas. Como é óbvio, a rede de núcleo que interliga uma cidade à outra é a mesma. As VPNs estabelecidas entre as sucursais permitem o envio de informação sem qualquer tipo de erros ou perdas. Neste cenário, pretendia-se também observar a troca de etiquetas MPLS realizada na rede de núcleo, o que foi facilmente conseguido e explicado. Verificou-se que a informação era muitas vezes enviada por um determinado percurso e recebida por outro. A primeira justificação avançada foi o facto dessa decisão ser baseada nos custos que foram atribuídos às interfaces dos routers. Neste caso, os custos dos percursos eram exactamente os mesmos, em ambos os sentidos. Depois de se alterarem os custos de algumas portas, verificou-se que alguns dos percursos deixavam de ser utilizados, o que

prova que a selecção dos percursos se baseia nos custos do protocolo de encaminhamento interno que está a ser utilizado.

Em suma, neste trabalho foram estudadas algumas características importantes do MPLS e referenciados alguns pormenores interessantes. No entanto, dada a extensão do tema muito ficou ainda por estudar.

Do ponto de vista pedagógico, creio que este trabalho no futuro poder vir a ajudar os alunos a perceber um pouco melhor como funciona o MPLS e a utilidade de alguns dos seus principais mecanismos.

Bibliografia

- J. M. B. Patrão, "Aspectos de dimensionamento de redes MPLS: optimização de nós e ligações," Universidade de Aveiro Tese de Mestrado, 2003.

- Network Working Group, "Constraint-Based LSP Setup using LDP," The Internet Society, 2002.

- E. Gimenez, R. Vieira, M. Cardoso, and G. Ferrari, "Engenharia de Tráfego nas Redes MPLS: Uma Análise Comparativa de Seu Desempenho em Função de Suas Diferentes Implementações," in *World Congress on Computer Science, Engineering and Techology Education*, São Paulo, 2006, pp. 570-574.

- I. Pepelnjak and J. Guichard, *MPLS and VPN Architectures*. Indianapolis, USA: Cisco Press, 2007.

- L. K. Chin, "Rede Privada Virtual - VPN," *Boletim bimestral sobre tecnologia de redes*, vol. 2, no. 8, Nov. 1998.

- IPSEC. Security Project at the TCM Laboratory - "IPSEC - Internet Protocol Security". [Online]. http://www.tml.tkk.fi/Tutkimus/IPSEC/ipsec.html

- Cisco. (2008, Aug.) Packet Tracer 5.0. [Online]. http://www.imakenews.com/ciscona-latamportuguese/e_article001168347.cfm?x=b11,0,w

- NS-2. (2009, Jul.) User Information. [Online]. http://nsnam.isi.edu/nsnam/index.php/User_Information

- OPNET. (2009) [Online]. http://www.opnet.com/

- Peopleware. (2009, Jan.) GNS3 0.6. [Online]. http://www.pplware.com/2009/01/08/gns3-06/

- E. Osborne and A. Simha, *Traffic Engineering with MPLS*. Indianapolis, USA, 2002.

- B. Davie and Y. Rekhter, *MPLS: Technology and Applications*. San Diego, USA: Academic Press, 2000.

- L. Lobo and U. Lakshman, *MPLS Configuration on Cisco IOS Software*. Indianapolis, USA: Cisco Press, 2005.

- H.-W. Choi and Y.-T. Kim, "Configuration Managements for BGP/MPLS VPN and DiffServ-aware-MPLS VPN," *KNOM Reveiw*, vol. 6, no. 2, pp. 80-89, 2004.

- D. Falsarella. (2008, Oct.) iMasters - "Implementação MPLS TE". [Online]. http://imasters.uol.com.br/artigo/10385/redes/implementacao_mpls_te/

- D. Maughan, M. Schertler, M. Schneider, and J. Turner. (1998, Mar.) Internet Security Association and Key Management Protocol (ISAKMP). [Online]. http://www.imib.med.tu-dresden.de/imib/Internet/Literatur/ISAKMP/draft-ietf-ipsec-isakmp-09.txt

- J. Pinheiro. (2006, Feb.) Projecto de Redes - "O MPLS em Redes de Computadores". [Online].

http://www.projetoderedes.com.br/artigos/artigo_mpls_em_redes.php

- TechNet-Microsoft. (2001, Sep.) Virtual Private Networking: An Overview. [Online]. http://technet.microsoft.com/en-us/library/bb742566.aspx

- J. Werner, "Tecnologias para Implantação de Redes Virtuais Privadas," in *Fórum Nacional sobre Segurança de Redes e Telecomunicações*, 1998.

- C. Goldani, "Multiprotocol Label Switching (MPLS)," Unicert, 2005.

- J. Ruela, "MPLS - Multiprotocol Label Switching," FEUP/DEEC/RBL, 2005.

- Projeto GIGA, "MPLS-Multiprocol Label Switching," UFRJ, 1996.

- Cisco, "Packet Tracer 5.0 Overview," Networking Academy, 2008.

- protocols.com. MPLS protocols family - MPLS|LDP|CR-LDP|RSVP-TE . [Online]. http://www.protocols.com/pbook/mpls.htm

- openmaniak.com. (2008, Mar.) IPERF - The Easy Tutorial. [Online]. http://openmaniak.com/iperf.php

- Wikipedia. (2009, Jul.) Network simulator. [Online]. http://en.wikipedia.org/wiki/Network_Simulator

- Manage Engine. (2009) QEngine. [Online]. http://www.manageengine.com/products/qengine/qengine-features.html

- M. Portnoi, "CR-LDP: ASPECTOS E FUNCIONAMENTO," Universidade Salvador – UNIFACS Tese de Mestrado, 2005.

Referências

[1] J. M. B. Patrão, "Aspectos de dimensionamento de redes MPLS: optimização de nós e ligações," Universidade de Aveiro Tese de Mestrado, 2003.

[2] Network Working Group, "Constraint-Based LSP Setup using LDP," The Internet Society, 2002.

[3] E. Gimenez, R. Vieira, M. Cardoso, and G. Ferrari, "Engenharia de Tráfego nas Redes MPLS: Uma Análise Comparativa de Seu Desempenho em Função de Suas Diferentes Implementações," in *World Congress on Computer Science, Engineering and Techology Education*, São Paulo, 2006, pp. 570-574.

[4] I. Pepelnjak and J. Guichard, *MPLS and VPN Architectures*. Indianapolis, USA: Cisco Press, 2007.

[5] L. K. Chin, "Rede Privada Virtual - VPN," *Boletim bimestral sobre tecnologia de redes*, vol. 2, no. 8, Nov. 1998.

[6] IPSEC. Security Project at the TCM Laboratory - "IPSEC - Internet Protocol Security". [Online]. http://www.tml.tkk.fi/Tutkimus/IPSEC/ipsec.html

[7] Cisco. (2008, Aug.) Packet Tracer 5.0. [Online]. http://www.imakenews.com/ciscona-latamportuguese/e_article001168347.cfm?x=b11,0,w

[8] NS-2. (2009, Jul.) User Information. [Online]. http://nsnam.isi.edu/nsnam/index.php/User_Information

[9] OPNET. (2009) [Online]. http://www.opnet.com/

[10] Peopleware. (2009, Jan.) GNS3 0.6. [Online]. http://www.pplware.com/2009/01/08/gns3-06/